



**André Filipe Lopes  
Augusto**

**Activação Automática de Serviços em redes  
MPLS e IP**





**André Filipe Lopes  
Augusto**

**Activação Automática de Serviços em redes  
MPLS e IP**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor Rui Valadas, Professor Associado do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, e co-orientação do Mestre Jorge Gonçalves, responsável pelo departamento de Sistemas de Suporte às Operações, da PT Inovação, S.A.



## **o júri**

### **presidente**

**Prof. Catedrático José Carlos da Silva Neves**

Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

### **arguente**

**Prof. Doutor Edmundo Heitor da Silva Monteiro**

Professor associado com Agregação do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

### **orientador**

**Prof. Doutor Rui Jorge Morais Tomaz Valadas**

Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro



## **agradecimentos**

A toda a minha família, em especial à minha esposa Clara pelo carinho, amor e em especial a todo o apoio prestado principalmente nas horas de “aperto”.

Gostaria de agradecer ao Engenheiro Jorge Gonçalves (PTInovação) pela oportunidade de realizar esta dissertação baseada no trabalho que estou a realizar.

Ao Professor Doutor Rui Valadas por ter aceite a orientação deste trabalho.

A toda a equipa de trabalho SSO4 da PT Inovação S.A., especialmente ao Alexandre, assim como aos colaboradores da Withus.

A todos os amigos que sempre estiveram presentes para dar moral, humor e apoio.

Finalmente quero agradecer a todas as outras pessoas que não mencionei aqui, mas que de uma maneira ou de outra contribuíram positivamente para o meu trabalho.

A todos: Obrigado.





## palavras-chave

NGOSS, Activação de Serviços, OSS/BSS, J2EE, eTOM, SID, XML/XSLT, OSS/J, TMForum

## resumo

A área das telecomunicações está constantemente a sofrer períodos de crescimento, baseados nas inovações tecnológicas que vão surgindo a todo o momento. Perante este crescimento, as empresas de telecomunicações debatem-se com a crescente competitividade do sector, o que as obriga a contínuas inovações de modo a aumentarem a sua oferta de serviços no mercado, mas também a uma contenção e mesmo redução de custos, no desenvolvimento dessas mesmas inovações. Perante tal situação, estas empresas precisam de estar aptas a desenvolver e alterar rapidamente os serviços que oferecem, incluindo também os processos de negócio associados e os sistemas de operação que os suportam. Estas exigências de negócio têm levado os prestadores de serviço a sistematizar os seus processos de negócio e a evoluir a arquitectura de interligação de sistemas OSS/BSS no sentido de permitir automatizar processos, incluindo-se neste campo os processos de provisão de serviços.

É neste contexto que surge o NGOSS - *New Generation Operations Systems and Software* como uma iniciativa do *TeleManagement Forum*. Esta iniciativa propõe princípios e modelos de negócio que incluem, entre outros, a automatização dos processos de provisão de serviços, conceito normalmente designado por *Flow Through Provisioning*, que visa disponibilizar duma forma rápida e eficiente os serviços subscritos pelos clientes, reduzindo ao mínimo a intervenção manual. O atingir deste objectivo requer a existência de uma infra-estrutura que facilite a interligação e a troca de informação entre os sistemas OSS/BSS.

Nesta tese pretende-se dar a conhecer esta iniciativa, demonstrando com um caso prático de activação de serviços desenvolvido pela PTInovação, que esta é uma solução viável no futuro das empresas de telecomunicações.



**keywords**

NGOSS, Service Activation, OSS/BSS, J2EE, eTOM, SID, XML/XSLT, OSS/J, TMForum

**abstract**

The area of telecommunications is constantly undergoing periods of growth, based on technological innovations that are emerging all the time. With this growth, the telecommunications companies are struggling with the growing competitiveness of the sector, which require them to be continuous innovative in order to increase their service offering to the market, but also a restraint and even cost savings in developing those innovations. Faced with this situation, these companies need to be able to quickly develop and modify the services they offer, including the business processes and associated operating systems that support them. These requirements have led to business service providers to systematize their business processes and evolve the architecture of OSS/BSS systems interconnection in order to automate processes, for example the processes of service provision.

In this context comes NGOSS - New Generation Operations Systems and Software as an initiative of the TeleManagement Forum. This initiative proposes principles and business models that include, among others, the automation of procedures for the provision of services, a concept usually referred to as Flow Through Provisioning, which aims to provide a quick and efficient way to subscribe services by clients, by minimizing technician manual intervention. Achieving this goal requires the existence of an infrastructure to facilitate the interconnection and exchange of information between OSS / BSS systems.

In this thesis, the NGOSS initiative will be presented, demonstrating with a service activation case study developed by PTInovação, that this is a viable solution in the telecommunications companies future.



# Índice

<b>ÍNDICE .....</b>	<b>I</b>
<b>LISTA DE FIGURAS .....</b>	<b>III</b>
<b>LISTA DE TABELAS .....</b>	<b>IV</b>
<b>LISTA DE ACRÓNIMOS .....</b>	<b>V</b>
<b>CAPÍTULO 1: INTRODUÇÃO .....</b>	<b>1</b>
1.1. MOTIVAÇÃO .....	1
1.2. OBJECTIVOS .....	1
1.3. ESTRUTURA DA DISSERTAÇÃO .....	2
1.4. NOTAÇÃO UTILIZADA .....	2
<b>CAPÍTULO 2: ORGANIZAÇÕES DE NORMALIZAÇÃO .....</b>	<b>5</b>
2.1. INTERNATIONAL TELECOMMUNICATION UNION .....	5
2.2. ITU TELECOMMUNICATION STANDARDIZATION SECTOR .....	6
2.3. INTERNET ENGINEERING TASK FORCE .....	6
2.4. TELMANAGEMENT FORUM .....	7
<b>CAPÍTULO 3: CONCEITOS DE REDES .....</b>	<b>9</b>
3.1. REDES ATM .....	9
3.2. <i>BACKBONE</i> .....	10
3.3. REDES IP .....	11
3.4. REDES MPLS .....	12
<b>CAPÍTULO 4: ARQUITECTURA DE OSS / NGOSS .....</b>	<b>15</b>
4.1. VISÃO GERAL .....	15
4.2. O QUE SÃO OSS? .....	16
4.3. OS NGOSS .....	19
4.3.1. <i>A arquitectura NGOSS</i> .....	21
4.3.2. <i>O modelo eTOM</i> .....	24
4.3.3. <i>O modelo SID</i> .....	27
4.4. IMPLEMENTAÇÃO PRÁTICA DE NGOSS – A INICIATIVA OSS/J .....	29
4.5. CLASSIFICAÇÃO FUNCIONAL DOS SISTEMAS OSS .....	34
4.6. CASOS DE APLICAÇÃO DE ARQUITECTURAS OSS .....	39
4.7. EVOLUÇÃO DOS SISTEMAS OSS E AS REDES IP .....	42
4.8. SURGIMENTO DE NOVOS SERVIÇOS .....	45
<b>CAPÍTULO 5: ACTIVAÇÃO AUTOMÁTICA DE SERVIÇOS - CASO PRÁTICO .....</b>	<b>47</b>
5.1. ARQUITECTURA DE REFERÊNCIA DE PROVISÃO DE SERVIÇOS .....	47
5.2. <i>NETWORK ACTIVATOR</i> .....	54
5.3. DEFINIÇÃO DE REQUISITOS .....	54
5.3.1. <i>Requisitos Funcionais</i> .....	56
5.3.2. <i>Requisitos Não Funcionais</i> .....	58
5.4. ARQUITECTURA DO SISTEMA .....	59
5.5. PERSPECTIVA FUNCIONAL .....	61
5.6. PERSPECTIVA LÓGICA .....	65

5.6.1. Descrição dos Módulos .....	67
5.7. PERSPECTIVA FÍSICA .....	71
5.8. PONTOS RELEVANTES DA ARQUITECTURA APRESENTADA .....	73
5.9. EXEMPLO DE CONFIGURAÇÃO DE SERVIÇO NUMA REDE MPLS .....	74
5.9.1. Configuração do Serviço .....	74
5.9.2. Configuração do Recurso .....	77
<b>CAPÍTULO 6: CONCLUSÕES E TRABALHO FUTURO.....</b>	<b>83</b>
<b>ANEXO I.....</b>	<b>87</b>
<b>ANEXO II .....</b>	<b>94</b>
<b>ANEXO III.....</b>	<b>96</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>103</b>

# Lista de Figuras

Figura 1 – Domínio MPLS .....	13
Figura 2 – Arquitectura TMN.....	18
Figura 3 – A arquitectura NGOSS.....	21
Figura 4 – Ciclo de vida da arquitectura NGOSS.....	23
Figura 5 – Mapa de processos eTOM.....	25
Figura 6 – Processos de nível 2 do Mapa de Estratégia, Infra-estrutura e Produto eTOM.....	26
Figura 7 – Processos de nível 2 do Mapa de Operações eTOM .....	27
Figura 8 – Vista de Negócio SID.....	29
Figura 9 – Evolução da integração de arquitecturas de gestão proposta pela iniciativa OSS/J. ....	31
Figura 10 – Arquitectura NGOSS implantada pela Telecom Italia para Serviços de banda larga.....	40
Figura 11 – Vista dos principais sistemas usados pela BT no seu projecto "Rede do século XXI" .....	41
Figura 12 – Arquitectura definida no projecto Catalyst "Managing IP Services with NGOSS". ....	42
Figura 13 – Arquitectura J2EE. ....	43
Figura 14 – Plano de desenvolvimento das APIs OSS/J. ....	44
Figura 15 – Estratégia da Empresa. ....	48
Figura 16 – Modelo de Referência do Processo de Provisão de Serviços [31].....	49
Figura 17 – Arquitectura de Referência. [29].....	52
Figura 18 – Provisão de Serviços – Arquitectura de Referência [29].....	53
Figura 19 – Modelo SID - Relação entre Produto, Serviço e Recurso. ....	55
Figura 20 – Conceito <i>Service Specification</i> .....	56
Figura 21 – Arquitectura do Sistema. ....	60
Figura 22 – Perspectiva Funcional do Sistema. ....	62
Figura 23 – <i>Use Cases</i> de Processamento de Pedidos. ....	63
Figura 24 – <i>Use Cases</i> de Configuração de Serviços. ....	64
Figura 25 – <i>Use Cases</i> de Configuração de <i>Plug-ins</i> . ....	64
Figura 26 – Estrutura de módulos do <i>Network Activator</i> . ....	65
Figura 27 – Entidades Relevantes do Sistema. ....	66
Figura 28 – Máquina de estados da entidade Ordem.....	69
Figura 29 – Composição do RFS rfsVplsAccess.....	75
Figura 30 – Representação do recurso lógico subInterfaceVlanEthernet. ....	76
Figura 31 – Estrutura de repositórios do sistema.....	77

# Lista de Tabelas

Tabela 1 – Requisitos Genéricos .....	57
Tabela 2 – Requisitos Externos .....	57
Tabela 3 – Requisitos da Arquitectura .....	58
Tabela 4 – Requisitos de Interfaces Externas .....	58
Tabela 5 – Requisitos Não Funcionais .....	59
Tabela 6 – Descrição das entidades relevantes do Sistema. ....	67



# Lista de Acrónimos

API	<i>Application Programming Interface</i>
ATM	<i>Asynchronous Transfer Mode</i>
B2B	<i>Business to Business</i>
BGP	<i>Border Gateway Protocol</i>
BSS	<i>Business Support Systems</i>
CBE	<i>Core Business Entities</i>
CCITT	<i>Comité Consultatif International Téléphonique et Télégraphique</i>
CFS	<i>Customer Facing Services</i>
CORBA	<i>Common Object Request Broker Architecture</i>
COTS	<i>Commercial, off-the-shelf</i>
CRM	<i>Customer Relationship Management</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
EJB	<i>Enterprise JavaBeans</i>
eTOM	<i>enhanced Telecom Operations Map</i>
EAI	<i>Enterprise Application Integration</i>
EMS	<i>Element Management System</i>
FAB	<i>Fulfillment, Assurance e Billing</i>
FTP	<i>Flow Through Provisioning</i>
IETF	<i>Internet Engineering Task Force</i>
IGMP	<i>Internet Group Management Protocol</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
ITU	<i>International Telecommunication Union</i>
ITU-T	<i>ITU Telecommunication Standardization Sector</i>
J2EE	<i>Java 2 Platform, Enterprise Edition</i>
LAN	<i>Local Area Network</i>
LR	<i>Logical Resource</i>
NA	<i>Network Activator</i>

NMF	<i>Network Management Forum</i>
MPLS	<i>Multi Protocol Label Switching</i>
NE	<i>Network Element</i>
NGOSS	<i>New Generation Operations Systems and Software</i>
ONU	<i>Organização das Nações Unidas</i>
OSI	<i>Open Systems Interconnection</i>
OSS	<i>Operational Support Systems</i>
OSS/J	<i>Operations Support System through Java Initiative</i>
OS&R	<i>Operations Support &amp; Readiness</i>
POP	<i>Points of Presence</i>
PPP	<i>Point-to-Point Protocol</i>
PSTN	<i>Public Switched Telephone Network</i>
QoS	<i>Quality of Service</i>
RFC	<i>Request for Comments</i>
RFS	<i>Resource Facing Service</i>
ROI	<i>Return on Investment</i>
SID	<i>Shared Information/Data</i>
SLA	<i>Service Level Agreement</i>
SONET	<i>Synchronous optical networking</i>
TCO	<i>Total Cost of Ownership</i>
TMForum	<i>TeleManagement Forum</i>
TMN	<i>Telecommunications Management Network</i>
TNA	<i>Technology Neutral Architecture</i>
UML	<i>Unified Modeling Language</i>
VoIP	<i>Voice over Internet Protocol</i>
VOD	<i>Video on Demand</i>
VPLS	<i>Virtual Private LAN Services</i>
VPN	<i>Virtual Private Networks</i>
WAN	<i>Wide Area Network</i>
WFM	<i>Work Force Management</i>
WSDL	<i>Web Service Description Language</i>
XML	<i>Extensible Markup Language</i>
XSLT	<i>Extensible Stylesheet Language Transformation</i>

# Capítulo 1: Introdução

## 1.1. Motivação

Os fornecedores de serviços são confrontados com desafios significativos para uma oferta de serviços *end-to-end*, envolvendo um conjunto díspar de actores/empresas. A gestão e a automatização de processos que envolvem várias empresas é essencial para assegurar um nível de excelência na prestação de serviços *end-to-end*. Assim a provisão de novos serviços de uma forma rápida e eficiente, suportada por um conjunto de processos e sistemas com um nível de integração elevado, é assumida como um factor de diferenciação num mercado extremamente competitivo.

A iniciativa NGOSS [1] promovida pelo TMForum [2] promove um conjunto de princípios para o desenvolvimento de uma infra-estrutura escalável e flexível para uma rápida oferta de serviços. Propõe a criação de processos de negócio através da utilização de componentes com contratos (interfaces) bem definidos, num ambiente distribuído. A definição formal de processos, repositório de processos ou motores de processos são algumas das peças necessárias para construir tal infra-estrutura.

O facto do trabalho que tenho vindo a realizar para a PTInovação estar inserido num projecto de Activação de Serviços fez com que esta fosse uma escolha óbvia para o tema desta tese, sendo uma mais-valia para a apresentação das arquitecturas e características destes tipos de sistemas, que se encaixam na iniciativa NGOSS.

## 1.2. Objectivos

O objectivo desta tese passa por compreender a importância de uma arquitectura de OSS para activação de serviços, descrever o processo de activação automática de serviços,

utilizando uma plataforma de activação de serviços e recursos e demonstrar uma activação automática de serviços EPL/EVPL e VPLS em redes MPLS;

### **1.3. Estrutura da Dissertação**

A presente tese está organizada do seguinte modo:

- Capítulo 2: é apresentada uma breve descrição de organizações de normalização e o seu contributo para a iniciativa NGOSS, da qual se destaca a activação de serviços, tema do trabalho prático apresentado nesta tese.
- Capítulo 3: são apresentados alguns conceitos de redes que servem de suporte a processos de suporte à operação, incluindo redes ATM, IP e MPLS.
- Capítulo 4: é apresentada a visão geral dos OSS/NGOSS, juntamente com a sua arquitectura, classificação, evolução e implementação prática.
- Capítulo 5: é apresentado um caso prático de activação de serviços, compreendendo a definição de requisitos, arquitectura, perspectivas lógicas, físicas e funcionais do sistema.
- Capítulo 6: são apresentadas as conclusões desta tese.

### **1.4. Notação utilizada**

Apesar de ter sido despendido algum esforço em contrário, o texto desta dissertação inclui por diversas vezes termos técnicos em língua inglesa cuja tradução não existe ou não exprime completamente o significado original. Esta situação deve-se fundamentalmente ao facto das publicações da especialidade e dos trabalhos levados a cabo pelas organizações de normalização utilizarem preferencialmente o inglês como veículo de difusão da informação. Nestes casos, os termos em questão são apresentados com caracteres *itálicos*.

Para evitar a repetição de longas expressões técnicas, que tornariam fastidiosa a leitura desta dissertação, são usados acrónimos ao longo de todo o texto. Além de ser apresentada no início deste documento, a correspondência entre cada termo técnico e o respectivo acrónimo é feita aquando da primeira ocorrência desse termo no texto.

Todas as referências bibliográficas a que se recorre ao longo deste texto são assinaladas entre parêntesis recto e apresentadas na secção final deste documento. Na mesma secção encontram-se também outras fontes de informação de carácter geral não referenciadas ao longo do texto que foram utilizadas globalmente durante a elaboração desta dissertação.



# Capítulo 2: Organizações de Normalização

É importante reconhecer que hoje em dia, nenhuma organização pode, por si só cuidar de todos os aspectos de negócio, o que implica uma grande colaboração entre todas as organizações envolvidas em determinadas áreas. A área das telecomunicações não foge à regra. São apresentadas de seguida algumas Organizações de Normalização e Fóruns que trabalham entre si e que são relevantes para o trabalho efectuado nesta tese.

## 2.1. International Telecommunication Union

A *International Telecommunication Union* (ITU) [3] é uma organização internacional que se destina a regular e a padronizar as radiofrequências e as telecomunicações internacionais. Esta organização, fundada originalmente como *International Telegraph Union* em Paris em 1865 é actualmente a organização internacional mais antiga do mundo, tendo como principais acções a regulação do espectro de ondas de rádio e a organização dos arranjos de interligações entre todos os países, permitindo ligações telefónicas internacionais. Devido à sua longevidade como organização internacional e o seu *status* como agência especializada da Organização das Nações Unidas (ONU), os padrões promovidos pela ITU possuem um grande valor de reconhecimento internacional sobre organizações que publicam especificações técnicas similares. Resumindo, a ITU é uma organização na qual os governos e o sector privado concentram esforços no sentido de coordenar a exploração de redes e serviços de telecomunicações, promovendo o seu desenvolvimento.

## 2.2. ITU Telecommunication Standardization Sector

*ITU Telecommunication Standardization Sector* (ITU-T) [4] coordena as normas para as telecomunicações, em nome da ITU. A ITU-T encontra-se baseada em Genebra, na Suíça. Esta surgiu em 1947 como um comité especializado da ONU e em 1956 ganhou *status* de organização, passando a ser denominada por *Comité Consultatif International Téléphonique et Télégraphique* (CCITT). Em 1993 passou a ser designada por ITU-T. Esta organização garante a eficiência e a produção de normas, abrangendo todos os campos das telecomunicações. Os padrões internacionais que são produzidos pela ITU são referidos como *Recommendations* (recomendações). Estas recomendações passarão a ser definitivas aquando da aprovação pelos membros da organização, e são empregues como referência no desenvolvimento de soluções tecnológicas envolvendo redes e telecomunicações. A ITU-T coopera o seu trabalho com outras organizações de normalização, por exemplo, a *International Organization for Standardization* (ISO) e a *Internet Engineering Task Force* (IETF) [5].

## 2.3. Internet Engineering Task Force

A *Internet Engineering Task Force* (IETF) [5] é uma organização que desenvolve e estimula padrões para a Internet, cooperando nomeadamente com as normas do protocolo TCP/IP e protocolos para a Internet. É uma organização com padrões abertos, e todos os participantes são voluntários, inclusive os seus trabalhos são normalmente realizados por empregadores ou patrocinadores. Está organizada em grupos de trabalho informais e grupos de discussão, cada um tratando de um tema específico. Cada grupo deve concluir os trabalhos sobre esse tema e depois é dissolvido. Os grupos de trabalho estão organizados em áreas como: Aplicações, Internet, Gestão de Operações, Aplicações e Infra-Estruturas em Tempo-Real, *Routing*, Segurança e Transportes. Os *Request for Comments* (RFC) são publicados pela IETF e descrevem métodos, comportamentos, ou inovações aplicadas ao funcionamento da Internet assim como a sistemas ligados a ela. Através da *Internet Society*, engenheiros e cientistas informáticos podem publicar os RFC, quer para revisão da comunidade em geral, ou simplesmente para transmitir novos conceitos ou informações. A IETF adopta algumas das propostas publicando como RFCs com normas a seguir.



Protocolos de internet como por exemplo DHCP, PPP e IGMP, entre outros, são exemplos de protocolos normalizados pela IETF.

## 2.4. TeleManagement Forum

O *TeleManagement Forum* (TMForum) [2], antes conhecido como *Network Management Forum* (NMF), é um consórcio mundial sem fins lucrativos com vários membros ligados à prestação de serviços na indústria das telecomunicações e fornecedores. Este consórcio visa o desenvolvimento de estratégias que ofereçam a liderança aos seus participantes na indústria das telecomunicações, desenvolvendo formas mais eficazes de melhoramento da gestão das redes e serviços públicos. O TMForum possui mais de 700 membros globais, em mais de 75 países, que se inserem nas indústrias de telecomunicações, cabo, média e Internet. As normas e *frameworks* desenvolvidas neste fórum, relacionadas como tema desta tese serão desenvolvidas mais à frente neste documento, como por exemplo o conceito *Operational Support Systems* (OSS), mais propriamente o *New Generation Operations Systems and Software* (NGOSS).



# Capítulo 3: Conceitos de Redes

As redes de telecomunicações ocupam hoje um papel muito importante na vida das pessoas. No entanto, estas tecnologias são relativamente recentes e encontram-se em constante evolução. Com o objectivo de oferecer sempre mais e melhores serviços aos clientes as redes de telecomunicações tornam-se cada vez mais complexas, o que faz com que a sua manutenção se torne cada vez mais complexa também.

De modo a gerirem de forma eficiente estas redes multi-tecnologia, as operadoras de telecomunicações evoluíram os seus sistemas de suporte à operação de modo a que estes implementassem os processos operacionais de forma independente das tecnologias de rede.

São apresentados de seguida alguns conceitos de redes que servem de suporte a estes processos de suporte à operação.

## 3.1. Redes ATM

*Asynchronous Transfer Mode* (ATM) [6] é uma arquitectura de rede de alta velocidade orientada á ligação e baseada na comutação de pacotes de dados (paradigma de comunicação de dados em que pacotes (a unidade de transferência de informação) são individualmente encaminhados entre nós da rede através de ligações de dados tipicamente partilhadas por outros nós) e não comutação de circuitos (que se caracteriza pela utilização permanente de circuitos durante toda a transmissão). As primeiras normas do ATM, ou Modo de Transferência Assíncrono (comutação e transmissão) surgiram nos anos 80 e foi desenhado como um protocolo de comunicação de alta velocidade que não depende de nenhuma topologia de rede específica. Usa uma tecnologia de comutação de células de alta velocidade que pode tratar tanto dados como vídeo e áudio em tempo real.

A introdução da fibra óptica veio revolucionar a tecnologia de transmissão. A fibra óptica permite comunicações a longa distância com taxas de erros muito inferiores às cablagens de cobre a curta distância. Este avanço provocou necessariamente uma inadequação da tecnologia de nível superior existente. Tradicionalmente uma das grandes preocupações era lidar de uma forma eficiente com taxas de erros relativamente elevadas. A definição do ATM teve em consideração este facto adoptando a técnica *Cell Relay*.

O protocolo ATM encapsula os dados em pacotes de tamanho fixo de 53 bytes (48 bytes de dados e 5 de cabeçalho). No ATM estes pacotes são denominados de células. Uma célula é análoga a um pacote de dados, à excepção que numa das células ATM nem sempre contém a informação de endereçamento de camada superior nem informação de controlo de pacote. Este tipo de transmissão de dados é escalável, permitindo que as suas células de 53 bytes possam ser transportadas de uma *Local Area Network* (LAN) para outra através de uma *Wide Area Network* (WAN). A velocidade do ATM começa em 25 Mbps, 51 Mbps, 155 Mbps e superiores. Estas velocidades podem ser atingidas com cabos de cobre ou fibra óptica (com a utilização exclusiva de cabo em fibra óptica pode-se atingir até 622.08 Mbps). Estas velocidades são possíveis porque o ATM foi desenhado para ser implementado por hardware em vez de software, sendo assim são conseguidas velocidades de processamento mais altas.

### **3.2. Backbone**

No contexto de redes de computadores, o *backbone* (traduzindo para português, espinha dorsal) designa o esquema de ligações centrais de um sistema mais amplo, tipicamente de elevado desempenho [7].

Por exemplo, os operadores de telecomunicações mantêm sistemas internos de elevadíssimo desempenho para comutar os diferentes tipos e fluxos de dados (voz, imagem, texto, etc.). Na Internet, numa rede à escala planetária, podem-se encontrar, hierarquicamente divididos, vários *backbones*: os de ligação intercontinental, que derivam nos *backbones* internacionais, que por sua vez derivam nos *backbones* nacionais. Neste nível encontram-se, tipicamente, várias empresas que exploram o acesso às telecomunicações sendo, portanto, consideradas a periferia do *backbone* nacional.

Em termos de composição, o *backbone* deve ser concebido com protocolos e interfaces apropriados ao débito que se pretende manter. Na periferia, desdobra-se o conceito de ponto de acesso, um por cada utilizador do sistema. É cada um dos pontos de acesso, vulgarmente referidos como *Points of Presence* (POP) que irão impor a velocidade total do *backbone*. Por exemplo, se um operador deseja fornecer 10 linhas de 1 Mbit com garantia de qualidade de serviço, o *backbone* terá de ser, obrigatoriamente, superior a 10 Mbit (fora uma margem especial de tolerância).

### 3.3. Redes IP

#### Protocolo IP

IP é um acrónimo para a expressão inglesa *Internet Protocol* (ou Protocolo de Internet), que é um protocolo usado entre duas ou mais máquinas em rede para encaminhamento dos dados.

Os dados numa rede IP são enviados em blocos referidos como pacotes ou datagramas (os termos são basicamente sinónimos no IP, sendo usados para os dados em diferentes locais nas camadas IP). Em particular, no IP nenhuma definição é necessária antes do *host* tentar enviar pacotes para um *host* com o qual não comunicou previamente.

O IP oferece um serviço de datagramas não confiável (também chamado de *best-effort*); ou seja, o pacote circula quase sem garantias na rede. O pacote pode chegar desordenado (comparado com outros pacotes enviados entre os mesmos *hosts*), podendo inclusive haver duplicação de pacotes na rede, podendo também ser perdidos por inteiro. Se a aplicação precisar de confiabilidade, esta é adicionada na camada de transporte. O serviço oferecido pelo IP é sem conexão. Portanto, cada datagrama IP é tratado como uma unidade independente que não possui nenhuma relação com qualquer outro datagrama. Como a comunicação é não-confiável, não sendo usados reconhecimentos *end-to-end* ou entre nós intermediários. Nenhum mecanismo de controlo de erros nos dados transmitidos é utilizado, excepto um *checksum* do cabeçalho que garante que as informações nele contidas, que são usadas pelos *gateways* para encaminhar os datagramas, estão correctas. Nenhum mecanismo de controlo de fluxo é empregue.

O IP é o elemento comum encontrado na internet pública dos dias de hoje. É descrito no RFC 791 [8] da IETF [5], que foi pela primeira vez publicado em Setembro de 1981. Este documento descreve o protocolo da camada de rede mais popular e actualmente em uso. Esta versão do protocolo é designada de versão 4, ou IPv4 [9]. O IPv6 tem endereçamento de origem e destino de 128 bits, oferecendo mais endereçamentos que os 32 bits do IPv4.

### 3.4. Redes MPLS

No contexto das redes de computadores e telecomunicações, o *Multi Protocol Label Switching* (MPLS) [10] é um mecanismo de transporte de dados pertencente à família das redes de comutação de pacotes. O principal objectivo do MPLS é a integração dos paradigmas de troca de rótulos (*labels*) com o nível tradicional de rotas em redes. Esta integração aumenta a eficiência no encaminhamento de dados e posiciona favoravelmente as redes para atender a demanda crescente por qualidade de serviços (QoS). O MPLS opera numa camada *Open Systems Interconnection* (OSI) [11] intermédia às definições tradicionais do *Layer 2* (Ligação de Dados) e *Layer 3* (Rede), pelo que se tornou recorrente ser referido como um protocolo de *Layer 2,5*. Foi concebido para permitir um serviço unificado de transporte de dados para aplicações baseadas quer em comutação de pacotes quer em circuitos. Pode ser usado para transportar vários tipos de tráfego, como pacotes IP, ATM, SONET ou mesmo *frames* Ethernet.

O MPLS permite que as operadoras de uma determinada rede tenham alto desempenho no desvio de tráfego de dados em situações críticas, tais como de falhas e congestionamentos. O MPLS permite assegurar que a transmissão de determinados pacotes tenha perdas ou atrasos imperceptíveis em função da capacidade de uma gestão de tráfego mais eficaz, possibilitando assim maior qualidade dos serviços e consequentemente maior confiabilidade. É normalmente utilizado em empresas de telecomunicações responsáveis por *backbones* que utilizam protocolos de roteamento *Border Gateway Protocol* (BGP4), *Quality of Service* (QoS) e *Service Level Agreement* (SLA) de modo a aumentar a sua credibilidade quanto à disponibilidade dos seus serviços.

Uma série de diferentes tecnologias foram previamente pensadas com idênticos objectivos do MPLS, tais como *Frame Relay* e ATM. As tecnologias MPLS têm vindo a

evoluir com os pontos fortes e fracos do ATM em mente. Muitos engenheiros de telecomunicações concordam que o ATM deve ser substituído por um protocolo que exija menos sobrecarga na rede e ao mesmo tempo preste serviços orientados à ligação com diferentes comprimentos dos pacotes.

O MPLS é uma tecnologia de encaminhamento de pacotes baseada em rótulos (*labels*) que funciona, basicamente, com a adição de um rótulo nos pacotes de tráfego à entrada do *backbone* e, a partir daí, todo o encaminhamento pelo *backbone* passa a ser feito com base neste rótulo. Comparativamente ao encaminhamento IP, o MPLS torna-se mais eficiente uma vez que dispensa a consulta das tabelas de *routing*.

Este protocolo permite a criação de *Virtual Private Networks* (VPN) garantindo um isolamento completo do tráfego com a criação de tabelas de *labels* (usadas para roteamento) exclusivas de cada VPN.

Além disso é possível realizar QoS dando prioridades diferentes a aplicações críticas, dando um tratamento diferenciado para o tráfego entre os diferentes pontos da VPN. A QoS cria as condições necessárias para o melhor uso dos recursos da rede, permitindo também o tráfego de voz e vídeo.

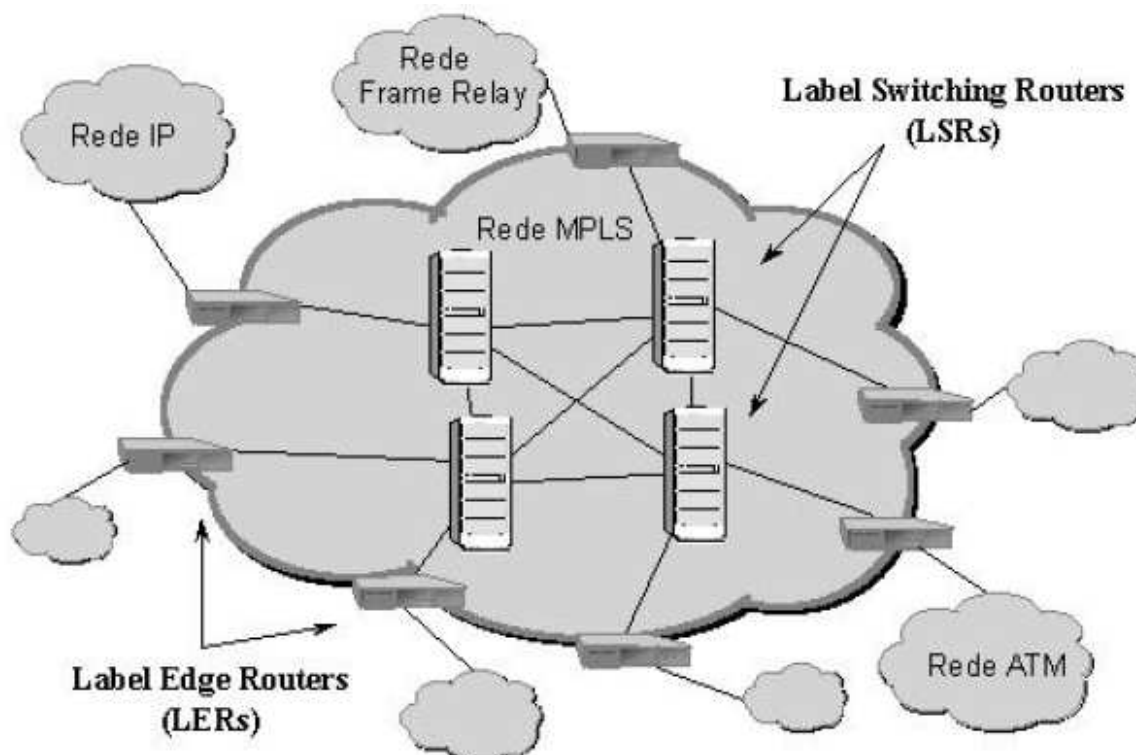


Figura 1 – Domínio MPLS





# Capítulo 4: Arquitectura de OSS / NGOSS

*Operations Support Systems* (OSS), ou em português, sistemas de suporte às operações em redes tornaram-se nos últimos anos, num pilar fundamental da viabilidade económica dos serviços disponibilizados pelas empresas de telecomunicações. Já não chega proporcionar uma gama de bons serviços aos clientes finais. É imperativo gerir esses serviços de uma forma bastante eficiente, melhorando e supervisionando cada um dos componentes da cadeia de elementos, que vão desde a venda do serviço em si, até à sua cobrança. Por estas razões, é importante conhecer as novas tendências em matéria de sistemas de suporte às operações, especialmente no que se refere aos sistemas de nova geração, os *New Generation Operations Systems and Software* (NGOSS) e a iniciativas como OSS/J.

## 4.1. Visão Geral

De acordo com pesquisas de mercado no sector das telecomunicações, o ano de 2005 foi o início de uma época onde os serviços de rede de nova geração e especialmente os serviços sobre IP representaram uma oportunidade de negócio multimilionária para os prestadores de serviços e para as operadoras de rede. Esse tempo ainda continua, e o aproveitamento ou não destas oportunidades baseia-se na capacidade que as empresas possuam para enfrentar a forte demanda nos cortes dos custos exigidos pelos clientes finais, assim como na capacidade de introduzir novos serviços adaptados às necessidades específicas dos seus clientes. Para uma arquitectura de sistemas de gestão flexível e multi-serviços, desenhada e optimizada para uma infra-estrutura IP, é crucial ter essa capacidade, oferecendo aos prestadores de serviços a oportunidade de contar com uma vantagem

competitiva de modo a ganhar quota de mercado à medida que o sector das telecomunicações avança para uma nova infra-estrutura de redes de nova geração.

É neste contexto que os OSS convertem-se numa peça chave para alcançar esses objectivos, permitindo uma maior eficiência na gestão de recursos e uma capacidade de resposta mais rápida perante a evolução contínua de processos e necessidades de negócio.

Esta tese apresenta de uma forma resumida os vários tipos de arquitectura OSS existentes onde se tenta determinar a sua aplicação, actual e futura, num contexto de redes e serviços IP, já que tudo indica que será esta tecnologia que se irá converter na mais dominante, se não já, nos próximos anos.

## 4.2. O que são OSS?

Antes de continuar esta tese, é necessário esclarecer o que são ao certo estes sistemas, genericamente conhecidos como *Operations Support Systems* (OSS) e apresentar a sua evolução até ao ponto de se converterem hoje em dia num factor chave para as políticas de corte de custos e para as estratégias de negócio das operadoras e prestadores de serviços dentro do mercado das telecomunicações.

Os sistemas de suporte às operações são aqueles que ajudam as empresas de telecomunicações a levar a cabo a gestão, monitorização e controlo das suas redes, bem como os serviços a correr sobre essas redes. Os sistemas de facturação, o apoio ao cliente, os serviços de gestão de clientes, a gestão da rede e equipamentos, estão incluídos na definição de OSS.

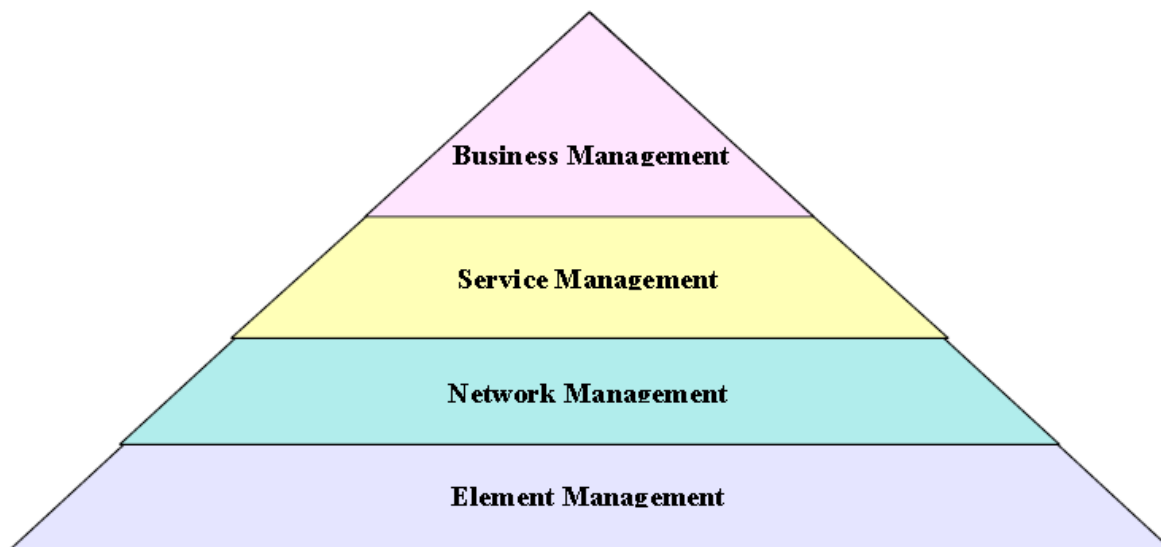
As considerações a tomar sobre a gestão de serviços inclui a manutenção dos novos clientes, a activação de serviços e a consequente segurança sobre esses serviços. A gestão destes serviços visa minimizar continuamente os custos da introdução de novos serviços, bem como a modificação dos serviços já existentes. Resumidamente, um OSS é um conjunto de sistemas que controla o fluxo completo de provisão de serviços, desde a activação até à facturação.

A arquitectura OSS foi definida pelo ITU-T no seu modelo *Telecommunications Management Network* (TMN) usado na gestão de redes das empresas de telecomunicações e serviços, como mostra a Figura 2. Os padrões estabelecidos pela TMN referem-se genericamente a um modelo de arquitectura funcional e especificações de protocolos de

comunicação criadas pelo ITU-T e outros organismos internacionais de padronização. O termo TMN em si refere-se a um conjunto de sistemas necessários ao suporte da gestão da rede e dos serviços de telecomunicações nas actividades de planeamento, provisionamento, instalação, manutenção, operação e administração.

As camadas definidas pela norma são:

- **Gestão de Negócio:** gere o conjunto dos processos empresariais, ou seja, coordena o planeamento de alto nível, as cobranças, as estratégias, decisões executivas e os contratos de negócios.
- **Gestão de Serviços:** utiliza informação disponibilizada pela camada de gestão de rede para gerir contratos de serviço de clientes existentes e potenciais, desde o aprovisionamento e a qualidade de serviço, até a gestão de falhas. Esta camada é também responsável pela interacção com os provedores de serviços e com outros domínios administrativos, mantendo dados estatísticos para garantir a qualidade do serviço prestado.
- **Gestão de Redes:** tem a visibilidade de toda a rede baseada em informações de elementos de rede disponibilizadas pela camada de gestão de elemento de rede. Esta camada coordena todas as actividades de rede e suporta as requisições da camada de gestão de serviços.
- **Gestão de Elementos de Rede:** gere cada elemento de rede. Esta camada possui sistemas de operação, e cada um dos quais é responsável pelas informações geridas de elementos de rede específicos. De uma forma geral, um gestor de elemento de rede é responsável por um subconjunto dos elementos de rede, gerindo os seus dados, actividades, registos, etc.



**Figura 2 – Arquitectura TMN.**

Originalmente, os OSS eram sistemas independentes, baseados em *mainframes*, e concebidos para ajudar as operadoras telefónicas nas suas tarefas diárias. Essencialmente, estes sistemas eram concebidos para efectuar os processos manuais, através dos quais uma rede telefónica era gerida de uma forma mais eficiente. Os provedores de serviços actuais, porém, necessitam de gerir um catálogo de serviços e tecnologias de rede muito mais complexa, isto se quiserem manter a sua competitividade. Com isto, as novas gerações de OSS estão a ser desenvolvidas actualmente para realizar a gestão da informação das empresas. Estes sistemas permitem que as informações de uma empresa sejam um recurso muito mais útil e acessível para a gestão de negócio, a provisão de serviços e realização de uma melhor gestão e atendimento a clientes, objectivo final dos operadores e prestadores de serviços, que são constantemente pressionados pela concorrência e se vêm forçados a alterar o seu modelo de negócios baseado na gestão optimizada dos seus recursos internos para um modelo onde a satisfação dos clientes é o principal motor dos seus processos de negócio.

De modo a alcançar um aumento efectivo no corte de custos nas operações e manutenção das infra-estruturas e aumentar o nível de satisfação do cliente perante as respostas que foram dadas aos seus requisitos e necessidades, torna-se evidente a necessidade de possuir uma estratégia clara dentro da área dos OSS e *Business Support Systems* (BSS) que permita, entre outros aspectos, uma automatização eficiente *end-to-end* dos processos de negócio. Esta automatização implica uma contínua automação entre os distintos sistemas de gestão que representarão, portanto, uma arquitectura perfeitamente coesa. Conforme descrito em [12] de uma forma mais pormenorizada, algumas das

iniciativas que vão surgindo visam a definição de uma solução que represente uma arquitectura de sistemas de gestão eficiente e flexível. De todas estas iniciativas, a que é mais amplamente aceite, é a arquitectura *New Generation Operations Systems and Software* (NGOSS), liderada pelo TeleManagement Forum (TMF) [2].

### 4.3. Os NGOSS

Os NGOSS fornecem um conjunto de especificações de referência para as soluções OSS e BSS de nova geração, e estão orientados ao modelo e automação dos processos de negócio. Para isto definem uma arquitectura de sistemas, modelos de dados e informação, normas e interfaces de integração entre todos os seus componentes especificando ainda uma metodologia de utilização para todos eles.

Em Maio de 2004 foram apresentados vários acordos entre o TMForum e dois dos fóruns que se encontram directamente ligados com as definições de um modelo de arquitectura de gestão. Estes dois fóruns são:

- O ITU [3], organização onde as operadoras possuem um papel predominante;
- A *Operations Support System through Java Initiative* (OSS/J) [13], liderada pela Sun e pela comunidade Java e que conta com o apoio de fornecedores de OSS como por exemplo, a MetaSolv e Telcordia, e de operadoras de telecomunicações tão importantes como a BT, Vodafone e Deutsche Telekom;

Por um lado, o acordo com a ITU é que esta irá considerar o modelo *enhanced Telecom Operations Map* (eTOM) (componente *core* da arquitectura NGOSS, que será descrito mais adiante), como uma recomendação oficial da ITU-T, dentro dos modelos de gestão de serviços incluídos na arquitectura de gestão de redes de telecomunicações (*Telecommunications Management Network* - TMN) que esta organização define. Deste modo pretende-se utilizar o modelo eTOM como uma ferramenta de consenso para a discussão e futuros acordos entre os prestadores de serviços, as operadoras de rede e os seus parceiros e fornecedores, o que contribuirá para alcançar uma visão geral e comum dos processos de negócio, e irá facilitar acordos entre organizações de outras áreas.

Por outro lado, a iniciativa OSS/J, um grupo de trabalho que se dedica ao desenvolvimento de interfaces de programação (APIs) *standard* para as distintas funcionalidades de OSS através de extensões da tecnologia J2EE e o TMF realizaram uma aliança [14] onde as APIs OSS/J se associam aos componentes NGOSS de uma forma mais generalizada, o modelo eTOM e o modelo *Shared Information/Data* (SID), a fim de assegurar a consistência entre uma aproximação centrada na obtenção de uma implementação real a partir do uso da linguagem de programação Java e as tecnologias J2EE e *WebServices*. O conjunto de APIs OSS/J consiste em especificações públicas e implementações de referência disponíveis em *open-source* e ferramentas de verificação utilizadas para a sua certificação, todas elas disponíveis de forma gratuita. A combinação das duas permite aos fornecedores e fabricantes implementar não apenas uma abordagem comum relativamente à informação e aos processos das tarefas de integração dos seus OSS e BSS, mas também poderem usar algumas APIs de implementação, disponíveis de forma gratuita para o desenvolvimento das suas soluções.

Por todas estas razões apresentadas, serão os princípios da arquitectura NGOSS em geral, e a definição de processos de negócio realizado pelo eTOM em particular, que servirão de premissas a seguir na gestão das redes actualmente existentes e serviços IP, cuja complexidade e natureza heterogénea fazem com que seja crítico dispor, de acordo com as directrizes do TMForum:

- Uma abordagem *top-down* que permita o fluxo completo dos processos de provisão de serviços das redes IP através de múltiplas camadas do modelo eTOM.
- Uma política de facturação e comprovação da qualidade oferecida, adaptada aos serviços a prestar.
- Uma arquitectura OSS que suporte:
  - Uma rápida alteração /melhoria do sistemas OSS, em resposta a mudanças de paradigmas de negócio que possam surgir;
  - Interfaces *standard* e comuns entre os diversos subsistemas OSS, que conduzam a uma fácil integração entre eles;
  - Suficiente capacidade em todos os componentes OSS para poderem interagir entre sistemas de vários fabricantes, e permitir a evolução a tempo do surgimento de novas exigências do negócio e avanços tecnológicos.

### 4.3.1. A arquitectura NGOSS

Não se pretende aqui descrever em detalhe os fundamentos da arquitectura NGOSS, mas sim dar umas ideias gerais, e entrar apenas em detalhe quando for feita alguma referência a alguns dos seus componentes básicos, tais como eTOM e SID, já que ajudam a definir os mapas dos sistemas que são responsáveis pelo trabalho de gestão e exploração de uma rede, e o modelo de dados que é necessário partilhar de modo a atingir os objectivos de eficiência e a coesão necessários para a optimização de processos de negócio.

NGOSS [15] é um marco de soluções orientadas ao negócio que especifica uma metodologia para a construção de componentes OSS. Para isso define uma arquitectura (ver Figura 3) para a gestão de redes de comunicação de nova geração (não apenas IP), cujos objectivos são:

- Reduzir o tempo de implementação de novos serviços (*Time to Market*).
- Reduzir os custos de integração.
- Reduzir os custos e os tempos de gestão.
- Facilitar a introdução de novas tecnologias.
- Suportar múltiplas tecnologias para apoiar a sua implementação (tecnologicamente neutra).

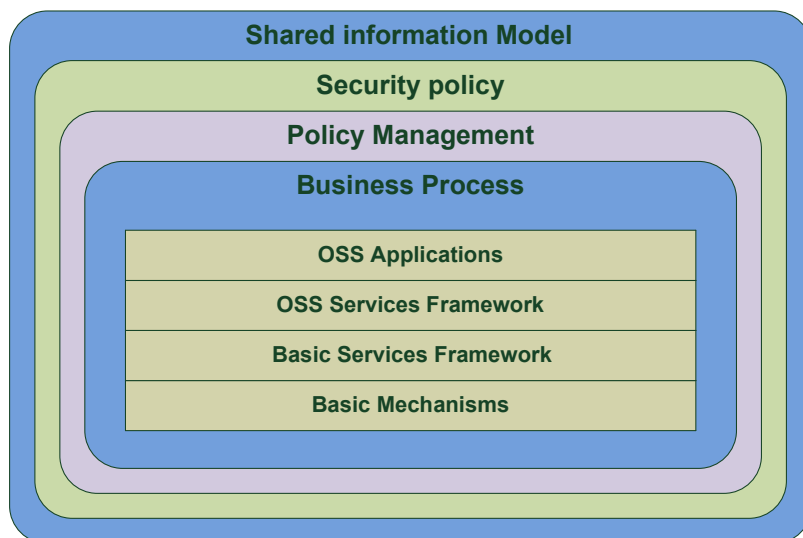


Figura 3 – A arquitectura NGOSS.

Para atingir estes objectivos, a arquitectura NGOSS rege-se por princípios que podem ser agrupados em dois tipos, de negócio e técnicos, dos quais se podem destacar os seguintes:

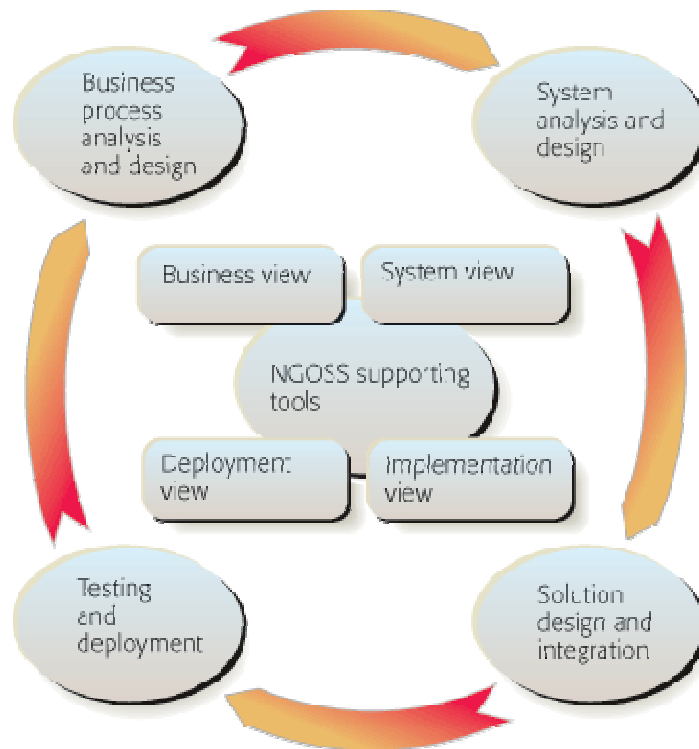
- É uma arquitectura baseada em componentes.
- Os serviços e as interfaces entre os sistemas para proporcionar estes serviços são definidos através de contractos.
- É utilizado um repositório comum de dados e de serviços, juntamente com um modelo de informação comum para todos os componentes.
- Os processos de fluxo de informação encontram-se separados.
- É uma arquitectura independente da implementação usada para o seu desenvolvimento, embora a sua execução seja feita através de várias tecnologias, como o CORBA, Java ou XML.
- É implementado com produtos comerciais (*Commercial, off-the-shelf* – COTS) [16] sempre que possível.

O processo de concepção, implementação e exploração da arquitectura NGOSS é definido por um ciclo de vida iterativo e orientado à obtenção da automação *end-to-end* dos processos de negócio através de uma correcta integração dos sistemas, conseguindo assim uma redução dos custos de operacionalização e uma maior eficácia nas respostas oferecidas aos clientes, de modo a aumentar o seu nível de satisfação. O ciclo de vida da arquitectura NGOSS baseia-se nas seguintes fases (ver Figura 4) [17]:

- **Fase de negócio:** nesta fase são definidos os mapas dos processos e sistemas e a integração de software. Para tal, o NGOSS utiliza o modelo eTOM, um esquema teórico que ajuda ao entendimento e especificação dos diferentes processos OSS e como estes se deverão relacionar entre si.
- **Fase de sistema:** nesta fase é especificado o fluxo de informação associada ao processo de negócio. Para tal é definido SID, modelo de informação genérico, com as principais entidades necessárias à gestão de redes e serviços. Com o SID são definidas, entre outras coisas, as funções de cada sistema.



- **Fase de implementação:** esta fase inclui a definição e implementação dos componentes, permitindo a escolha da tecnologia que seja considerada a mais adequada para cada um deles, e para a sua integração.
- **Fase de instalação:** nesta fase é colocada em prática o resultado das fases anteriores.



**Figura 4 – Ciclo de vida da arquitectura NGOSS.**

A arquitectura NGOSS possui outros conceitos importantes na sua especificação, como é o caso da *Technology Neutral Architecture* (TNA), que orienta e gere as actividades corporativas que levam ao desenvolvimento de modelos UML, que são formais e independente da tecnologia utilizada para os serviços definidos pelo eTOM e SID, permitindo assim, através da definição de contratos, a análise, a concepção, a implementação e instalação de soluções baseadas em NGOSS que sejam abertas e distribuídas. A arquitectura NGOSS define, além disso, uma série de procedimentos de verificação e certificação que permitem validar as normas definidas na implementação da arquitectura de gestão.

Nos capítulos seguintes serão descritos os conceitos eTOM e SID, que são fundamentais para a definição de uma arquitectura de gestão que permite a um operador ou

prestador de serviços de telecomunicações alcançar os seus objectivos de eficiência e de redução custos, e alcançar um alto grau de satisfação dos seus clientes.

#### **4.3.2. O modelo eTOM**

O eTOM, ou *enhanced Telecom Operations Map* [18], é um modelo de processos de negócio para os operadores de rede, prestadores de serviços e restantes intervenientes do sector das telecomunicações. Descreve todos os processos empresariais necessários a prestadores de serviços e analisa-os segundo diferentes níveis de detalhe, de acordo com a sua importância e prioridade para o negócio. Serve de modelo para a gestão de processos e fornece um ponto de referência neutro para os processos internos de reengenharia, bem como para os acordos e alianças com outros prestadores, estabelecendo uma linguagem de negócios e fundamentos para o desenvolvimento e integração dos sistemas OSS e BSS.

O modelo eTOM está, como já descrito anteriormente, muito ligado ao NGOSS, pois define aspectos-chave do ciclo de vida da fase negócio, oferecendo uma visão externa das funcionalidades suportadas pelas actividades de negócio. Além disso, os processos, fluxos e informações do eTOM são considerados requisitos para entrada na segunda fase do ciclo de vida da arquitectura NGOSS, a fase de sistema.

O eTOM também oferece suporte para, além de operadores e prestadores de serviços, aos gestores de sistemas, definindo os limites possíveis de componentes de software e as funções exigidas por estes, juntamente com os conceitos de entrada e saída que devem ser suportados por todos os produtos, e tudo isso, usando uma linguagem comum com os clientes, ou seja, com as operadoras e prestadores de serviços.

A *framework* de processos de negócio do eTOM caracteriza um ambiente completo de uma empresa prestadora de serviços. Esta *framework* começa na definição dos processos de negócio da empresa, agrupados de acordo com as suas características. É definida da maneira mais genérica possível de modo a que seja independente de qualquer organização, tecnologia e serviço, e podendo oferecer desta maneira um suporte global. Como se pode ver na Figura 5, num primeiro nível, o eTOM divide-se em 3 áreas principais do processo:

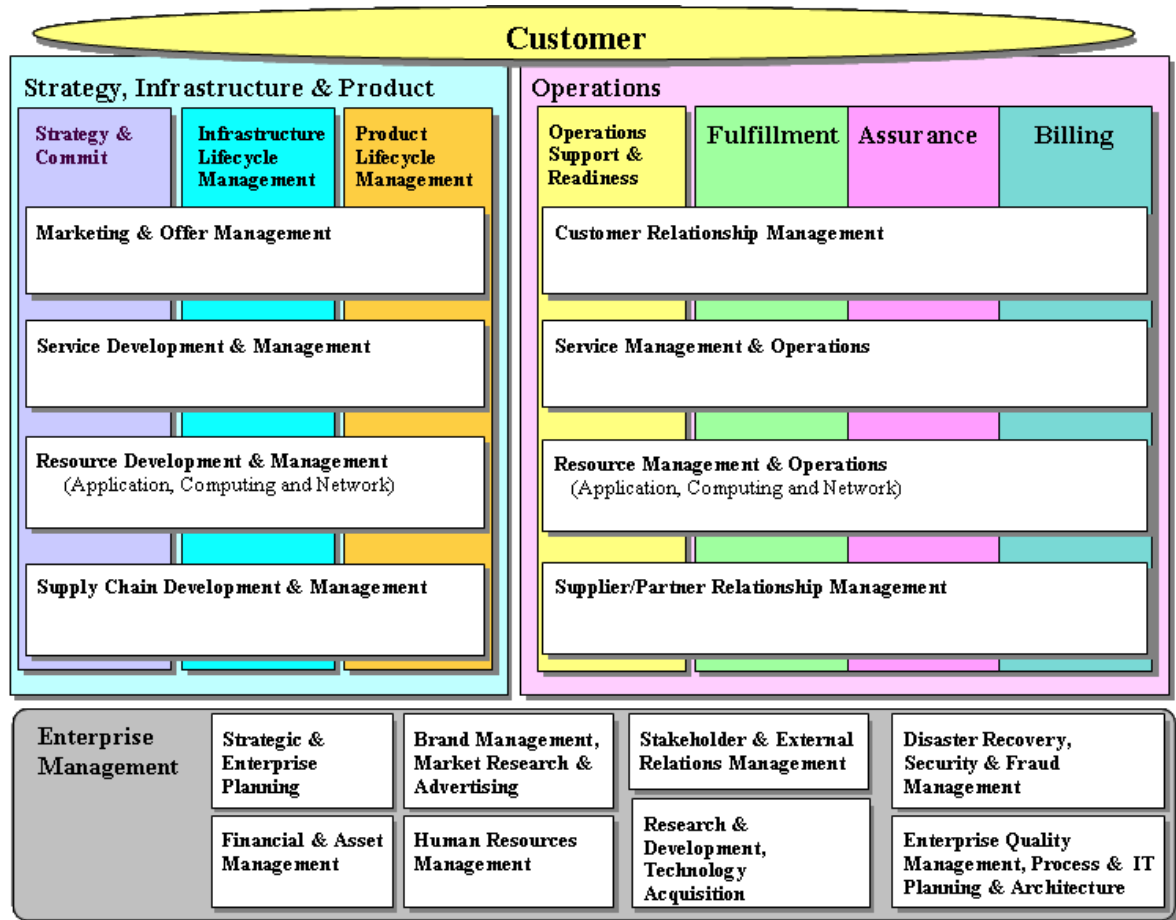


Figura 5 – Mapa de processos eTOM.

1. A área de Estratégia, Infra-estrutura e Produto, que abrange o planejamento e gestão do ciclo de vida dos processos de negócio.
2. A área de Operações, que abrange o núcleo da gestão operacional.
3. A área de Gestão da empresa, responsável pela gestão corporativa.

A estrutura dos processos no eTOM utiliza um conceito de decomposição hierárquica, pelo que os processos de negócio da empresa são progressivamente decompostos em níveis sucessivos, conforme apresentado na Figura 6 e Figura 7, onde se mostram os sub-processos da área de estratégia, infra-estruturas e produtos, e operações, respectivamente. Nesta sucessão de níveis são apresentadas as descrições dos processos, assim como as entradas e saídas de cada um dos processos, e qualquer outro elemento relevante para o processo.

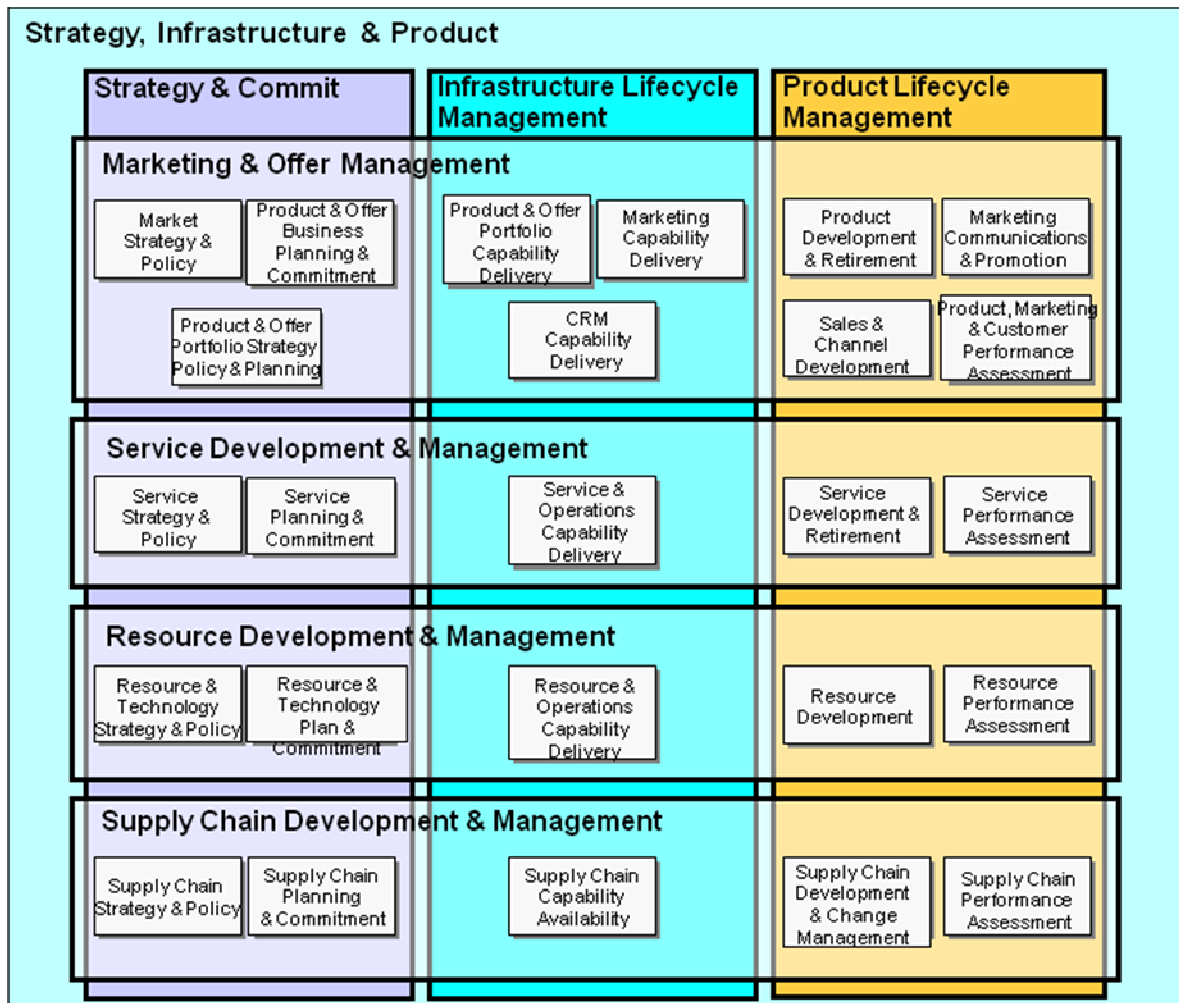


Figura 6 – Processos de nível 2 do Mapa de Estratégia, Infra-estrutura e Produto eTOM.

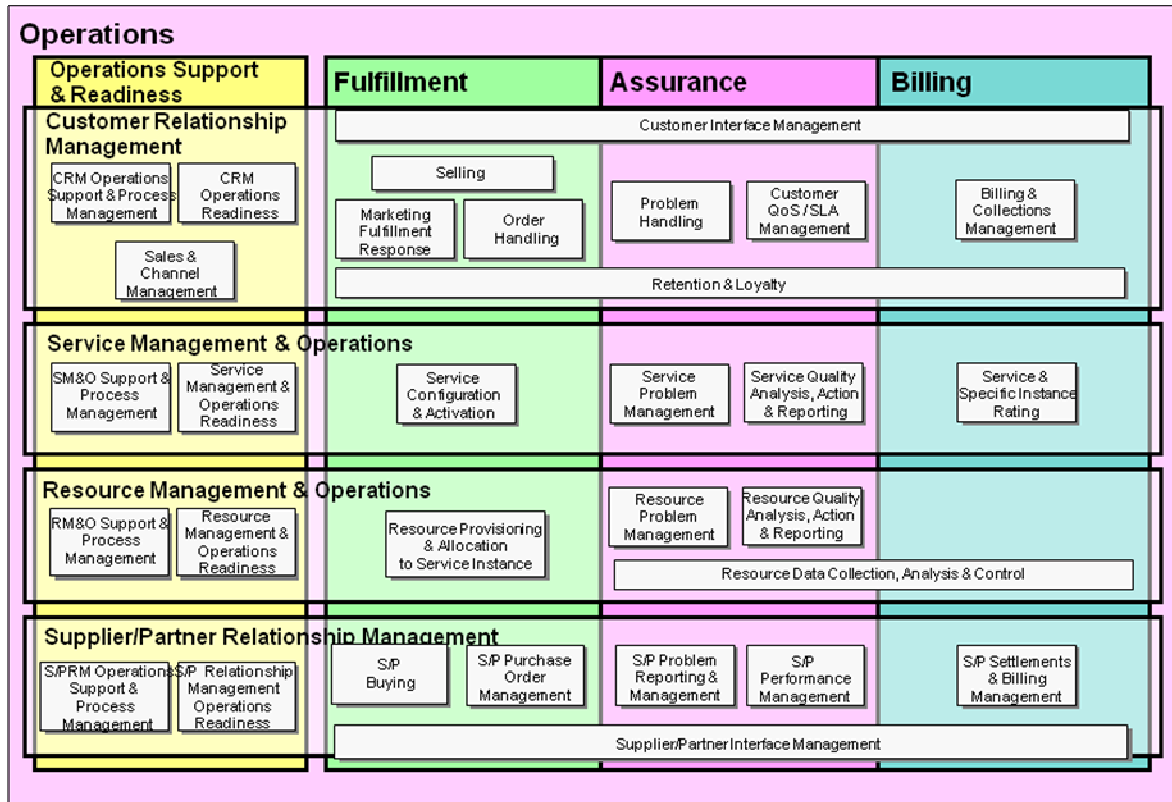


Figura 7 – Processos de nível 2 do Mapa de Operações eTOM

### 4.3.3. O modelo SID

SID ou *Shared Information/Data* [19] é um modelo de informação independente da plataforma, linguagem ou protocolo que seja utilizado para a sua implementação. Este é um modelo único que dá lugar a múltiplos modelos de dados, já que reflecte diferentes necessidades de gestão.

Por que razão é importante possuir um modelo partilhado de informações e dados? Para responder a esta questão basta olhar-mos para o estado actual dos sistemas de gestão de diversos operadores e prestadores de serviços, onde em muitas ocasiões existe informação replicada e existem diferentes terminologias associadas a diferentes grupos de operação responsáveis pelo funcionamento de cada sistema e das suas distintas necessidades. Esta replicação frequente gera ineficácia, aumentando os custos operativos. Com um modelo partilhado existe uma forma *standard* de estruturação, definição e implementação da informação e comportamentos dos processos de negócio, ao mesmo tempo permitindo a reutilização dos investimentos realizados em tecnologia da

informação, uma vez que evita a re-implementação dos mesmos modelos ao serem adicionados novos sistemas à arquitectura de gestão da rede.

O modelo de informação SID inclui conceitos tais como:

- Elementos de interesse, denominados entidades.
- Relações entre as entidades, denominadas de associações.
- Características das entidades, ou seja, atributos.
- Descrição do modo de trabalho das entidades, pelo que se definem métodos que representam o seu comportamento e as suas limitações, que restringem a sua funcionalidade, e que definem também as possíveis colaborações entre entidades.

As entidades que estão definidas numa primeira aproximação ao SID são aquelas que podem ser consideradas comuns a todos os operadores e prestadores de serviços. Estas entidades genéricas são as relativas a Produto, Cliente, Serviço, Mercado ou Vendas, Recursos, Fornecedores ou Parceiros, assim como as entidades Comuns de Negócio. Tal como o eTOM, o SID é constituído por vários quadros ou vistas, como a vista de Negócio (ver Figura 8) e a vista de Sistema, e define também um processo iterativo, pelo que em níveis sucessivos se vão incluindo ou completando todos os conceitos anteriores, como os atributos, os métodos, etc.

Quando SID é utilizado em combinação com os processos de negócio e com a descrição das actividades eTOM, então é criada uma ponte entre grupos de negócios e os grupos de tecnologia da informação, proporcionando definições compreensíveis aos grupos de negócio. Estas definições são suficientemente rigorosas para serem utilizadas pelos produtores de software.

O SID já está a ser utilizado pelos fabricantes de software, como é o caso da MetaSolv, Amdocs ou Intelliden, assim como pelos prestadores de serviços, tais como a BT [20] ou Telstra. Além disso, uma série de grupos associados à definição de arquitecturas de gestão estão actualmente a ponderar a sua utilização.

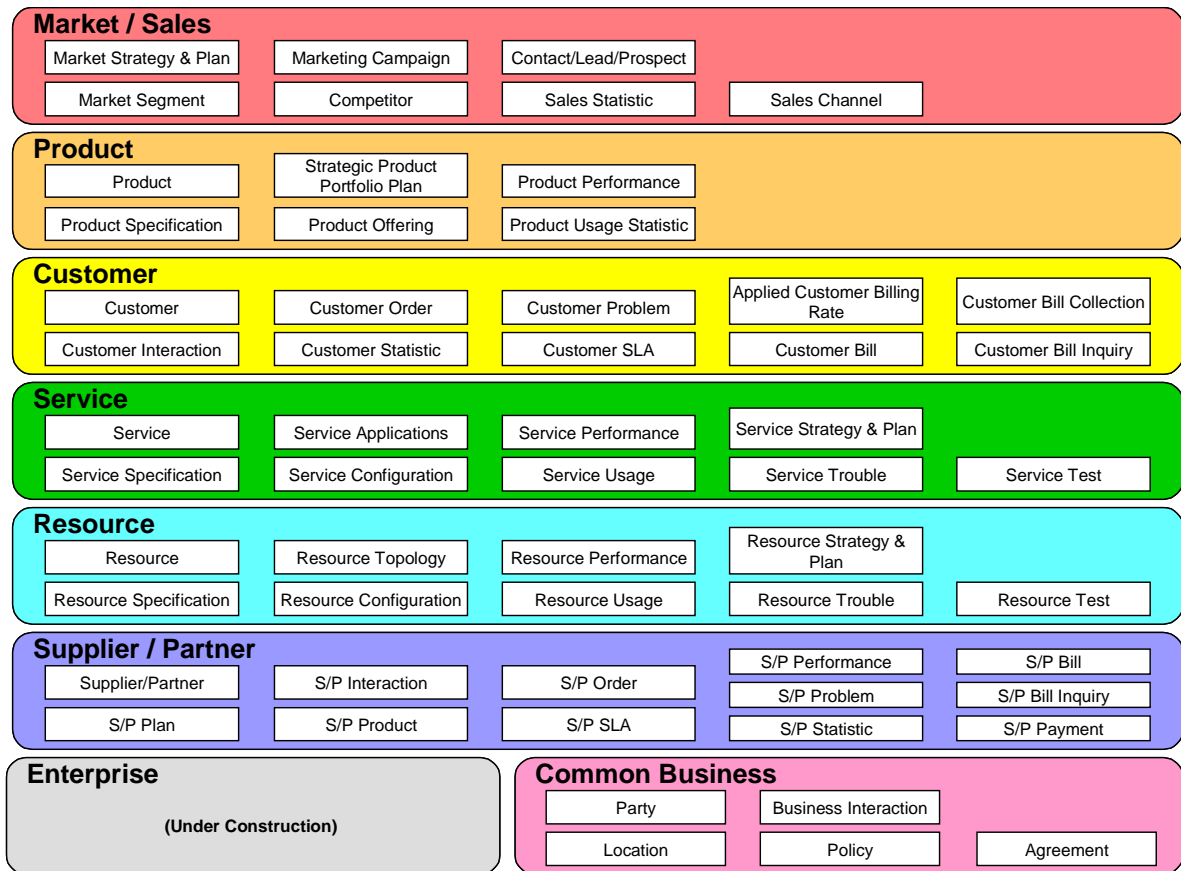


Figura 8 – Vista de Negócio SID.

#### 4.4. Implementação prática de NGOSS – A iniciativa OSS/J

Conforme descrito até agora, vimos que é necessária a definição de uma arquitetura de gestão de redes e serviços, não só para a tecnologia IP, mas para qualquer outra, e a iniciativa que apresenta melhores fundamentos teóricos e que se apresenta mais bem posicionada, é a NGOSS. No entanto, todos os conceitos apresentados até agora não passaram de um plano teórico, pelo que é necessário ir mais longe e optar por uma tecnologia que ponha em prática todas estas recomendações e os pressupostos a fim de se alcançar um mapa final de sistemas avançados e com um grau de ligação que permita à empresa atingir os objectivos desejados em termos de eficiência, redução de custos e satisfação do cliente.

A iniciativa OSS/J [14] define e implementa um conjunto de APIs baseadas em linguagens Java e tecnologia J2EE que permitem facilitar a implementação dos serviços *end-to-end* sobre as redes de nova geração e aproveitar a convergência entre as soluções de telecomunicações e soluções baseadas na internet. O objectivo desta iniciativa é proporcionar à indústria das telecomunicações soluções de operação e gestão de serviços e redes actuais e de nova geração.

A origem desta iniciativa surge quando se verificava que, na busca de soluções para a implementação de arquitecturas de gestão que seguissem as normas recomendadas por diversas organizações, como o TMF ou da ITU, as despesas inerentes à correcta integração entre todos os componentes superavam os custos de aquisição das próprias ferramentas disponíveis. Além disso, devido a cada vez mais exigentes prazos de execução, os integradores eram obrigados a duplicar código e métodos, e em múltiplas ocasiões, o resultado final não era nem é satisfatório, uma vez que requer um alto custo de manutenção e evolução e não apresenta as cotas de eficiência desejadas. Estes aspectos estão a ser levados em conta hoje em dia, perante as novas exigências do sector, bem como a aparição de equipamentos, serviços e tecnologias a ter em conta e, portanto, a integrar nos mapas de sistemas dos operadores.

A actual situação exige novos requisitos provenientes dos novos serviços, e a coexistência de novos sistemas, na maior parte das vezes baseados em produtos COTS [16] e sistemas tradicionais.

Na Figura 9 podemos ver a evolução das arquitecturas dos sistemas de gestão e operação das operadoras e prestadoras de serviços até agora, juntamente com a evolução proposta pela iniciativa OSS/J. A abordagem tradicional para a integração de sistemas tem consistido sempre na criação de interfaces ponto a ponto entre os diversos sistemas. Isto levava, por vezes a situações de extrema complexidade e de manutenção custosa, ao mesmo tempo mostrando-se bastante ineficiente em certas ocasiões. Esta situação foi melhorada com o surgimento de certos elementos de comunicação, *buses*, denominados como *Enterprise Application Integration* (EAI) [21], que oferecem uma plataforma comum para a comunicação entre os diversos actores, permitindo aumentar a eficácia e reduzir a replicação de código. No entanto, esta abordagem, ainda que melhore bastante os inconvenientes apresentados pela abordagem anterior, não evita que os custos de integração permaneçam elevados, nem que a substituição de uma forma simples dos



elementos que não alcancem o rendimento satisfatório nos processos de exploração da rede.

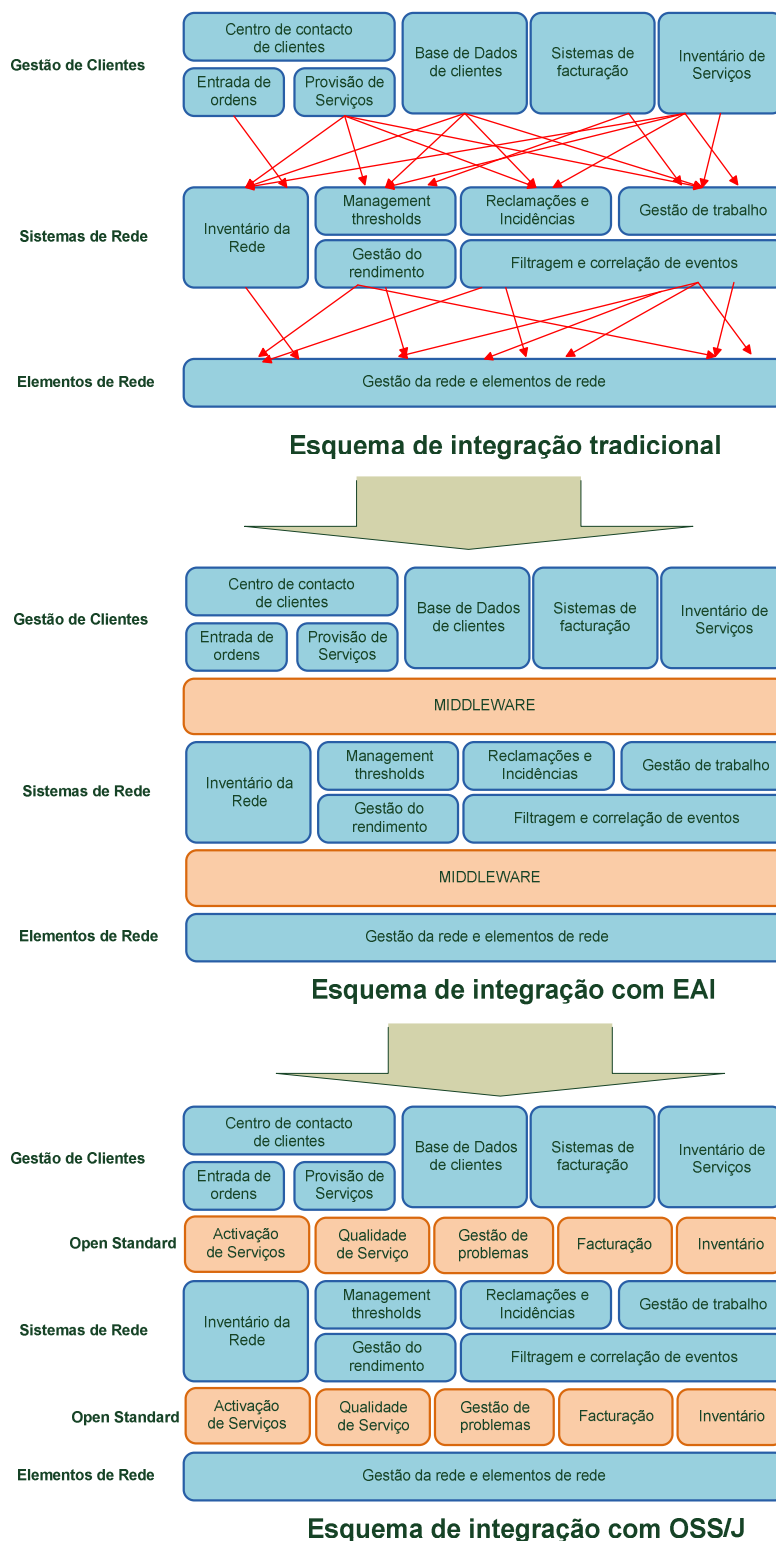


Figura 9 – Evolução da integração de arquitecturas de gestão proposta pela iniciativa OSS/J.

A evolução que propõem a iniciativa OSS/J é o passo seguinte, para que os custos de integração sejam minimizados, pois os sistemas contarão com APIs *standard*, e por consequência comuns, resultando numa redução dos custos de manutenção, num incremento da eficácia, resultando num *time-to-market* muito mais rápido no desenvolvimento e implantação dos novos serviços.

Conforme podemos ver na Figura 9, a iniciativa OSS/J centra-se na concepção e criação de APIs entre componentes, permitindo então uma total liberdade no momento da implementação de tais componentes. Isto faz com que a OSS/J conte com bastantes apoios entre fornecedores de sistemas OSS (destacando a Amdocs, MetaSolv e Telcordia) e entre os principais operadores (por exemplo, a BT, Vodafone e AT & T Wireless, mesmo a realizar testes baseados no conceito desta iniciativa, com excelentes resultados).

As características das APIs propostas pela iniciativa OSS/J podem-se resumir aos seguintes pontos:

- São genéricas, isto é, independente do serviço, de tecnologia, etc.
- Suportam consultas e metadados.
- São extensíveis.
- Reduzem a complexidade de integração.
- Proporcionam um ambiente aberto para o desenvolvimento, uma vez que as especificações, a implementação de referência e testes de compatibilidade são gratuitos e estão disponíveis a toda a comunidade.
- Fomentam a reutilização de código, incentivando o reaproveitamento de código, respeitando investimentos já realizados.
- Permite a troca de componentes sem custos adicionais de integração, pois se os componentes apresentam este tipo de interfaces bem definidas, então estaremos numa situação de *plug and work*.

Neste momento, há já uma série de APIs publicadas ou em revisão, juntamente com muitas outras já planeadas ou à espera de alguém que lidere a sua implementação. Neste sentido, a empresa Sun [22], criadora da iniciativa, bem como a comunidade Java, estão em constante procura de parceiros que possam levar a cabo este trabalho. As APIs publicados até agora abrangem um conjunto de processos ou áreas de negócio, tais como:

- Uma API comum a todos os OSS.
- A activação de serviços.
- A qualidade de serviço.
- A gestão de problemas.
- A facturação.
- O inventário.

Como discutido anteriormente, existe um acordo de colaboração entre a iniciativa OSS/J e o TMF, assinado em Maio de 2004, para que o OSS/J e o NGOSS caminhem lado a lado na busca e evolução das suas propostas. A ideia vem desde cedo, uma vez que, e embora as origens destas iniciativas pudessem ser consideradas contrárias, na prática, verificou-se que os princípios que se guiam são praticamente os mesmos, de modo que o resultado foi, de um ponto de vista teórico e tecnologicamente neutro do NGOSS, e de um ponto de vista mais prático e baseado na tecnologia Java do OSS/J, que ambas as iniciativas revelaram-se complementares.

Existem muitos pontos comuns entre OSS/J e NGOSS, uma vez que as APIs actuais do OSS/J cobrem parte da funcionalidade do modelo eTOM, e o plano de evolução de novas APIs vai continuar nesse sentido, até ser alcançada uma correspondência completa com todos os processos de negócio.

As semelhanças não são apenas na associação entre APIs e a funcionalidade dos processos eTOM, indo mais além, uma vez que a arquitectura J2EE, o fundamento do OSS/J suporta praticamente de forma directa os princípios de desenho do NGOSS. Nesse sentido, e como exemplo dessas associações, o OSS/J utiliza também um modelo de informação partilhada baseado nas chamadas *Core Business Entities* (CBE), que modelam os requisitos de informação das distintas APIs e que supõem uma implementação prática do modelo SID. Outro exemplo é a semelhança encontrada nos contratos entre os processos definidos pelo NGOSS e pelas interfaces Java utilizadas pelo OSS/J. Por último, os testes de compatibilidade e certificação de ambas as iniciativas também possuem bastantes semelhanças. Existem outros pontos semelhantes, que ainda estão em processo de definição e verificação. Todos estes aspectos farão que, à medida que as iniciativas avancem em colaboração, o OSS/J se converta numa implementação prática e completa da arquitectura NGOSS.

## 4.5. Classificação funcional dos sistemas OSS

Dentro do vasto campo dos sistemas OSS é necessário ter em conta o tipo de produtos que levam à prática todos os princípios atrás mencionados. Este capítulo da associação dos grupos funcionais de soluções com os processos de negócio definidos pelo mapa de operações do eTOM. Esta tarefa é sensível e de execução demorada, uma vez que os produtos desenvolvidos pelos fornecedores de sistemas OSS comerciais abrangem em variadas ocasiões mais do que um dos processos definidos por este modelo. Esta questão é importante na hora de escolher quais os componentes que irão formar a arquitectura dos sistemas OSS de uma empresa, já que poderá existir sobreposição de funcionalidades oferecidas pelos sistemas que se vai implementar. No projecto Catalyst, desenvolvido pelo TMForum para a gestão de serviços IP, estabeleceram-se variadas associações entre a funcionalidade fornecida pelo OSS e os processos utilizados no mapa de operações do eTOM, como se indica em [23] e [24].

A classificação dos sistemas OSS pode ser feita em função da funcionalidade fornecida. Neste sentido, todas as categorias e âmbitos de operação existentes dão lugar a um número elevado de empresas fornecedoras de software para OSS. No entanto, e apesar do facto de que actualmente o número de fabricantes de sistemas OSS ser elevado, existe uma tendência para a sua redução, especialmente agora que a grande maioria das operadoras converge a sua estrutura numa única rede. Nesta situação, nem todos os fornecedores serão capazes de sobreviver, principalmente aqueles que não estão bem posicionados no domínio das futuras redes de nova geração. Tendo em conta os sistemas OSS, com funcionalidade próximas à gestão de negócio, estes podem-se classificar do seguinte modo:

- **Sistema de facturação**

Os sistemas de *billing* ou facturação são responsáveis pela produção de facturas, a tempo de serem enviadas aos clientes, e correctamente preenchidas. Estes sistemas são igualmente responsáveis pelo processamento de pagamentos e as respostas aos clientes, pela investigação de reclamações, relativos a facturas, pela resolução de problemas de facturação, pela informação dada ao cliente sobre o estado da facturação e pelo suporte aos serviços pré e pós pago.

- **Sistema de *revenue assurance* e gestão de fraudes**

Estes sistemas verificam a correcta facturação, detectando e identificando possíveis usos não autorizados da rede do fornecedor de serviços. Alguns produtos nesta área podem determinar as actividades dos clientes e identificar padrões invulgares, da mesma forma que as companhias de cartões de crédito determinam padrões dos gastos dos seus clientes.

- **Sistemas de mediação**

Estes sistemas encarregam-se basicamente do processamento das transacções de alto nível. Juntam informações provenientes de vários elementos em rede em registos específicos, que são posteriormente despejadas noutros sistemas OSS, principalmente sistemas de facturação e gestão de eventos da rede. Os sistemas de mediação surgem geralmente ligados ao sistema de facturação, já que são fundamentais para o correcto funcionamento do operador. Por esse motivo, muitos dos fabricantes de sistemas de facturação incluem o seu próprio módulo mediação.

- **Sistemas de atenção ao Cliente (*Customer Relationship Management* - CRM)**

Os sistemas de CRM incluem todas as funcionalidades relacionadas com a aquisição e manutenção de clientes, e pode cobrir uma ampla gama de processos dentro um prestador de serviços, que vão desde o suporte ao cliente até ao *marketing*, mas mantendo sempre como objectivo central do seu funcionamento, a gestão das relações com o cliente.

Esta classificação está mais próxima do trabalho de gestão de negócio do operador ou prestador de serviço do que do trabalho de operações, exploração e manutenção das redes, podendo ser considerada com um sistema BSS. Devido aos trabalhos que desempenham, este tipo de sistemas não estão limitados a uma determinada tecnologia em particular. Nesse sentido, podemos efectuar uma segunda classificação de sistemas que são mais específicos para os trabalhos de provisão, gestão e manutenção das redes de telecomunicações. Podemos classificar assim:

- **Sistemas de gestão elementos multi-fabricante**

Os fabricantes de equipamentos devem fornecer aos prestadores de serviços e operadores de rede meios para a gestão remota dos seus equipamentos desde a sua instalação. No

mínimo, é necessário configurar placas e recolher alarmística. Por esta razão, praticamente todas os equipamentos da rede comunicam com os sistemas fornecidos pelos seus fabricantes. Este tipo de sistema é denominado de *Element Management System* (EMS), e geralmente empregam múltiplos protocolos e interfaces na comunicação com os equipamentos que gerem. Praticamente todos os fabricantes de hardware possuem o seu próprio EMS. Neste grupo também se encontram empresas que são apenas fornecedores de software, empresas que desenvolvem os seus sistemas através de acordos com os diversos fabricantes de equipamentos, e os fabricantes de hardware que, a partir dos EMS de gestão do seu próprio equipamento, estenderam a sua funcionalidade para se poderem ocupar dos equipamentos fabricados por outras empresas.

- **Sistemas de comunicação e de integração (*middleware*)**

Estes sistemas não são específicos no mercado das telecomunicações, embora a aplicação de alguns deles estão praticamente confinadas a este mercado. Apesar disso, esses sistemas desempenham um papel fundamental na configuração das arquitecturas de gestão dos sistemas OSS, servindo de *bus* de comunicações entre os sistemas OSS distintos de uma operadora. Englobam-se aqui os sistemas EAI usados actualmente na integração das arquitecturas OSS (como descrito no capítulo dedicado a OSS/J), e as plataformas J2EE, visto que os servidores aplicativos contam também múltiplas facilidades de comunicação e de integração entre sistemas.

- **Sistema de gestão de recursos (sistemas de inventários)**

Os sistemas de gestão de recursos, mais conhecidos como sistemas de inventário (embora esta definição tende a sugerir que não passem de bases de dados dos elementos físicos da rede, quando, na realidade abrangem muito mais), são hoje em dia parte fundamental da arquitectura do sistema OSS de gestão de um operador ou um prestador de serviços. Estes sistemas mantêm o inventário tanto dos recursos físicos como dos recursos lógicos. No caso do inventário físico da rede, estes sistemas possuem a informação relativa aos equipamentos de rede, a sua descrição, as suas placas, os *sites*, os portos, as ligações, etc. Estes sistemas permitem associar equipamentos instalados na rede com serviços prestados sobre eles e determinar por exemplo que clientes poderão ficar afectados pela falha de um dos elementos da rede, utilizando as relações configuradas entre equipamentos, serviços e clientes. Outra missão fundamental destes sistemas é determinar o nível capacidade e

ocupação da rede num determinado momento, servindo de ferramenta essencial no trabalho de planeamento e optimização desta.

Actualmente, a principal tendência quanto à utilização de sistemas de inventário é que estes sistemas sejam o núcleo central de qualquer arquitectura de gestão OSS, com ênfase no conceito de inventário centralizado de redes e serviços. Desta forma, os restantes sistemas OSS terão de se basear nas informações relativas aos equipamentos, clientes e serviços, com base no inventário para a realização das suas actividades de operação, até mesmo ao ponto destes sistemas serem os coordenadores dos processos de provisão de serviços. Com estas abordagens, a necessidade de crescimento, a flexibilidade e desempenho que estes sistemas têm para oferecer são ainda maiores, fazendo com que a maioria deles opte pela tecnologia J2EE como veículo para o seu desenvolvimento.

Na categoria dos sistemas de inventário poder-se-iam incluir uma gama de produtos que, sem ser em produtos de inventário propriamente ditos, mas que tendem a colaborar directamente com eles, até ao ponto dos sistemas de inventário incorporarem módulos para a realização dessas tarefas. São estes os sistemas de descoberta e de reconciliação, conforme descrito no ponto seguinte.

- **Sistemas de descoberta e de reconciliação**

Para uma operadora de telecomunicações e para um prestador de serviços é essencial que os seus sistemas de inventário apresentem uma informação fidedigna do estado da rede e dos serviços associados, ao mesmo tempo que sistemas de gestão distintos mantenham uma correcta sincronização entre a informação que partilham, em certas ocasiões, a replicam. Há, portanto, duas tarefas fundamentais que ajudam nessa sincronização de informação:

- a. A tarefa de descoberta de equipamentos e configurações de rede, que pode ser executada de forma automática ou gerida por um operador, e que permite obter informações da rede que irá ser adicionada ao sistema de inventário, de uma forma directa, ou após serem realizados testes de controlo em buscas de divergências que possam existir.
- b. A tarefa reconciliação, que consiste na comparação de dados de diversas fontes, de modo a serem resolvidas discrepâncias entre elas e decisões relevantes para que os sistemas reflectam com exactidão o estado das redes e serviços.

- **Sistemas de gestão de falhas**

Estes sistemas recolhem e apresentam alarmes e eventos que são obtidos a partir de sondagens feitas aos equipamentos de rede ou aos gestores dos equipamentos da rede. Por vezes, estes sistemas permitem aos operadores ligar-se a elementos de rede específicos, a fim de verificar se existe qualquer outra informação relevante que possa determinar a causa do envio de alarmes. Muitos destes sistemas executam também, de uma forma agendada, funções de monitorização do desempenho dos equipamentos de rede, embora esta tarefa seja trabalho para outro grupo funcional desta classificação.

- **Sistema de monitorização de prestação.**

Estes sistemas monitorizam aplicações, equipamentos ou funções de rede. Por exemplo, numa rede IP, poderiam analisar a latência e as perdas de pacotes. Existem sistemas concebidos para monitorizar tecnologias específicas, por exemplo, o ATM ou *Frame Relay*.

- **Sistemas de gestão de serviços**

Com base em informações fornecidas pelos sistemas de gestão de falhas e monitorização de desempenho, estes sistemas de obtêm uma imagem das prestações oferecidas pelos serviços, baseada na percepção do cliente e nos sistemas de gestão da rede. Este tipo de sistemas OSS exige a definição de padrões de desempenho. São usados em conjunto com *Service Level Agreements* (SLA) e proporcionam informação necessária para determinar se o prestador de serviços cumpre os níveis de qualidade negociados com os clientes.

- **Sistemas de activação de equipamentos e serviços.**

Estes sistemas, assim como o sistema do próximo tópico, podem ser denominados de *service fulfillment*, que não são mais do que a combinação de todos os processos implicados na implementação de uma ordem de serviço e a sua provisão ao cliente. Os sistemas de activação são responsáveis pela configuração dos equipamentos de rede de modo a que o serviço seja disponibilizado para um determinado cliente. Esta tarefa poderia implicar a deslocação de um técnico e a instalação de equipamentos. Quando esta instalação de equipamento não é necessária, ou já foi realizada, os sistemas de activação



actuais ligam-se directamente aos equipamentos ou aos serviços EMS de modo a enviar os comandos necessários de configuração para que o serviço fique disponível para os clientes.

- **Sistemas de provisão de serviços**

Estes sistemas também são incluídos no grupo de sistemas de *service fulfillment*. A sua missão consiste em especificar quais os equipamentos, sistemas e partes da rede são necessários para o serviço. Também localizam certos recursos, como a largura de banda na rede de transporte. Portanto, as tarefas de provisão estão bastante relacionadas com as funções de desenho e engenharia e com os sistemas de gestão dos recursos ou de inventário. Os sistemas de provisão geralmente incluem uma ferramenta de gestão e concepção de *workflows* para gerir os processos automáticos e manuais necessários ao bom funcionamento das tarefas relacionadas com a provisão de uma ordem de serviço. Isto não implica que qualquer tipo de ordem tenha de ser integrada nestes sistemas, já que a integração necessária para a construção de novos fluxos de trabalho de provisão, quando se instalam novos serviços, requer um acompanhamento próximo, até mesmo equivalente ao desenvolvimento ferramentas proprietárias.

Existem muitos outros tipos de sistemas que não foram incluídos na presente classificação apresentada, mas são igualmente necessários, e que dão suporte ao trabalho de exploração, gestão e manutenção das infra-estruturas dos operadores e prestadores de serviços. São por exemplo os sistemas que incluem funcionalidades como o apoio à planificação e engenharia, ou sistemas de teste que permitem comprovar o funcionamento das redes, os sistemas de segurança, sistemas de *backup*, etc.

## **4.6. Casos de aplicação de arquitecturas OSS**

Uma vez apresentada a lista de princípios defendida pela arquitectura NGOSS e a classificação dos sistemas OSS, será útil saber se, na verdade, já que existem os princípios teóricos e as ferramentas para implementação de uma arquitectura de sistemas OSS, exista alguém que a ponha em prática. De facto, esse passo já foi dado por variadas operadoras, existindo vários casos de implantação.

Estas operadoras começam por realizar a fase de implantação, com a definição das arquitecturas de alguns dos sistemas OSS que são regidos pelos princípios do NGOSS e pela utilização de sistemas comerciais. O primeiro caso de aplicação apresentado é o da Telecom Italia, operadora pioneira na implementação de uma arquitectura de gestão NGOSS, inicialmente dedicada à prestação de serviços de banda larga e actualmente em fase de desenvolvimento de outras tecnologias (ver Figura 10) [25]. Nesta operadora, o inventário Xng de Granite, o sistema de EAI da Tibco e o sistema de provisão e activação da Syndesis, foram os produtos seleccionados que mais se destacam. O projecto de implantação da arquitectura NGOSS por parte da Telecom Itália foi internacionalmente reconhecido em diversos fóruns, como exemplo a seguir por outros operadores e prestadores de serviço.

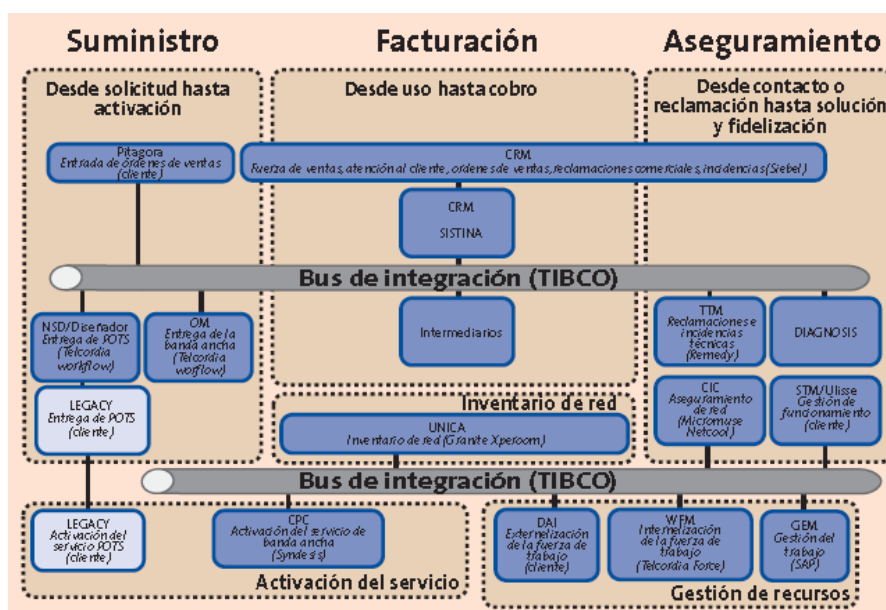
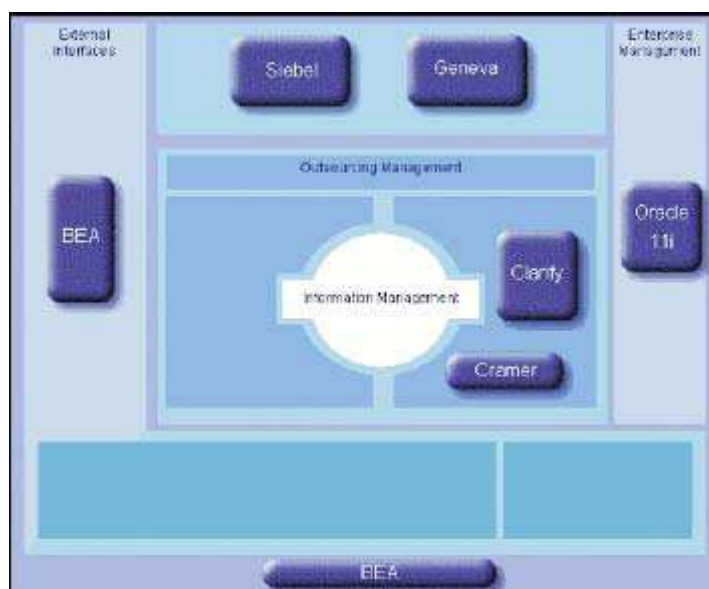


Figura 10 – Arquitectura NGOSS implantada pela Telecom Italia para Serviços de banda larga.

Um segundo caso de implantação de arquitecturas de sistemas OSS baseado em produtos comerciais pode ser encontrado na BT, assente num mega projecto denominado “Rede do século XXI”, que consistia na migração de todas as suas redes e serviços para a tecnologia IP, e a criação de uma arquitectura de sistemas OSS regida pelos princípios do NGOSS (ver Figura 11).

Além destas duas operadoras, que são importantes a nível europeu, muitas outras também realizaram o seu processo de implantação deste tipo de arquitecturas. Podem ser

mencionados outros casos, que não estão directamente associados com operadores ou prestadores de serviços IP, como é o caso da Korea Telekomk, na Ásia e a Covad, nos Estados Unidos da América. Estes dois exemplos estão a usar implementações das arquitecturas OSS baseadas nos princípios dos OSS/J, a primeira para a rede NGN e a segunda para a sua rede de banda larga. A Covad baseou-se principalmente nos sistemas fornecidos pela empresa NetCracker na construção da sua arquitectura OSS [26].



**Figura 11 – Vista dos principais sistemas usados pela BT no seu projecto "Rede do século XXI"**

No âmbito do TMForum realizam-se pequenos projectos que permitam concretizar os princípios de NGOSS, sendo um deles o projecto Catalyst "*Managing IP Services with NGOSS*", onde se desenvolvem arquitecturas de gestão para a provisão de serviços em redes privadas virtuais e em redes de voz sobre IP (VoIP), monitorização do rendimento da rede desenhada e a implementação de uma política de facturação centralizada. (ver a Figura 12).

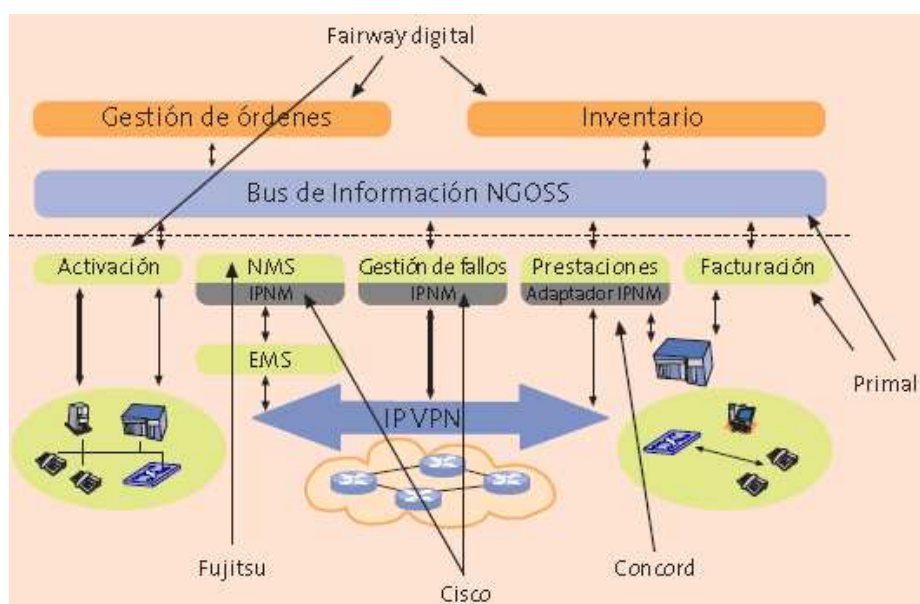


Figura 12 – Arquitectura definida no projecto Catalyst "Managing IP Services with NGOSS".

## 4.7. Evolução dos sistemas OSS e as redes IP

Como já foi descrito em capítulos anteriores, o caminho que os operadores e prestadores de serviços parecem estar a seguir actualmente no sentido da convergência das suas infra-estruturas e da utilização de tecnologia IP, requer a adopção de arquitecturas de gestão, cujos componentes, os sistemas OSS, estejam perfeitamente integrados entre si, de modo a alcançar níveis elevados de eficiência e rápida capacidade de resposta, com custos operacionais e custos de manutenção que não sejam elevados.

Neste sentido, a arquitectura NGOSS parece adaptar-se perfeitamente a essas necessidades. Foi inclusive comentado que o NGOSS é uma especificação neutra, uma vez que não indica as plataformas ou tecnologias que deverão ser usadas na sua implementação. Os sistemas OSS têm de garantir níveis muito elevados em termos de flexibilidade, capacidade de crescimento e rendimento, pelo que a tecnologia utilizada para a sua implementação deverá ser coerente com estas características.

Actualmente, a maioria das tendências do desenvolvimento de grandes sistemas parecem conduzir à adopção da plataforma J2EE para garantir a execução de óptimos níveis, em alguns aspectos, tais como mencionados anteriormente. Neste sentido, a semelhança entre os princípios da arquitectura J2EE e NGOSS torna ainda mais evidente

que poderá ser esta a tecnologia a adoptar pelos fornecedores de sistemas OSS, algo que já foi sendo feito, estando actualmente em fase de evolução dos sistemas de provisão, inventário e activação.

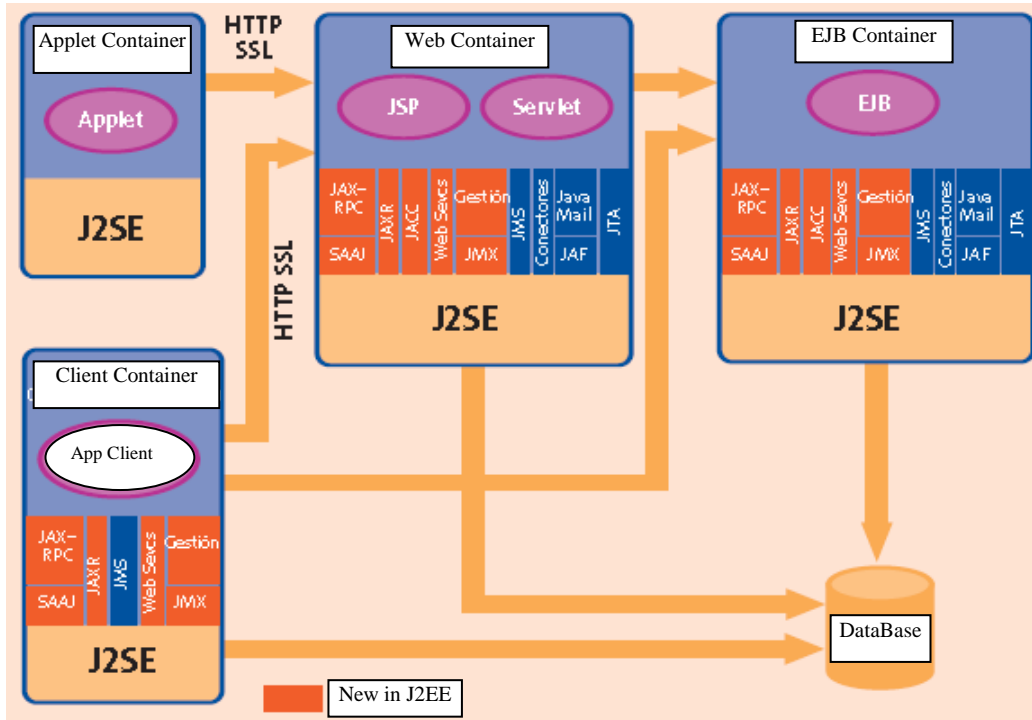


Figura 13 – Arquitectura J2EE.

Na verdade, as características que apresenta a plataforma J2EE enquanto arquitectura distribuída, com mecanismos de comunicação (EJBs), gestão de transacções, mecanismos que asseguram a capacidade de crescimento (*containers*) e modularidade, bem como as capacidades inerentes dos componentes J2EE, como a flexibilidade e reutilização, fazem com que aplicada ao desenvolvimento de sistemas OSS, se torne nada mais nada menos que uma representação real dos componentes NGOSS. O passo seguinte, desde que os sistemas OSS sejam implementados sobre esta tecnologia J2EE, seria a adopção da iniciativa OSS/J como veículo de integração entre os distintos componentes de uma arquitectura de sistemas OSS, uma vez que as APIs OSS/J não são mais que extensões de J2EE. Isso torna necessária a evolução das APIs previstas no plano de desenvolvimento OSS/J, de modo a ser possível a continuação da associação dos processos eTOM com estas APIs. (ver Figura 14).

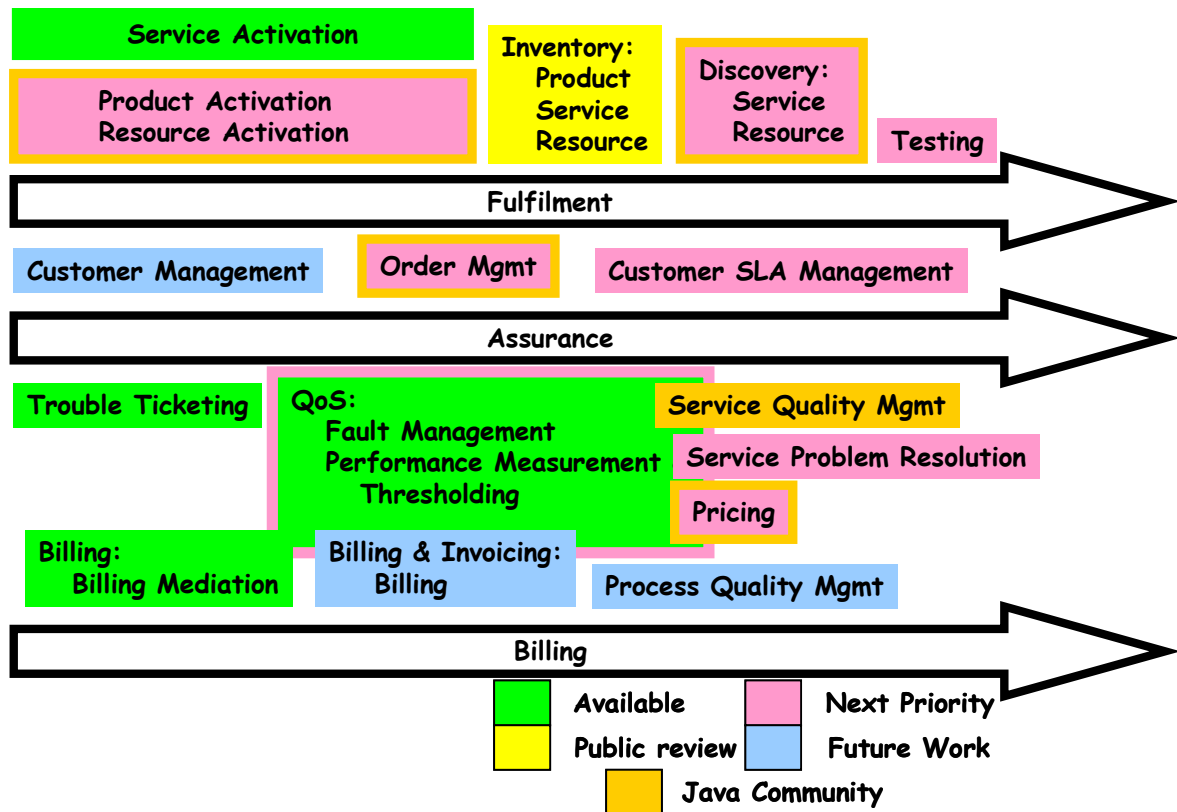


Figura 14 – Plano de desenvolvimento das APIs OSS/J.

Neste ponto é importante indicar que a iniciativa OSS/J não baseia o seu desenvolvimento apenas na implementação de novas APIs, mas também baseia o desenvolvimento no comportamento dinâmico de evolução do J2EE.

A adopção definitiva desta iniciativa, de modo a confirmar todas as expectativas levantadas, marcaria um avanço fundamental na obtenção de um dos factores mais importantes que se apresenta perante a implementação dos sistemas OSS: a redução dos custos da integração. Segundo estimativas dos operadores envolvidos nesta iniciativa, entre os quais se encontram a BT, Vodafone, Covad e a Deutsche Telekom, entre outras, as poupanças podem oscilar entre os 20 e 70 por cento, valores estes realmente importantes se tivermos em conta que a estimativa do crescimento do mercado de OSS será de uns 20 por cento por ano nos próximos anos.

## 4.8. Surgimento de novos serviços

Os novos serviços que vão surgindo como consequência da evolução das infra-estruturas de rede para uma única rede convergente, exigem que a arquitectura OSS dos operadores e prestadores de serviços seja extremamente flexível e apresente uma capacidade de resposta praticamente imediata. Um destes serviços é o serviço de voz sobre IP (VoIP), que oferece uma grande flexibilidade e transparência, bem como a possibilidade de suportar serviços avançados de nova geração que integrem voz e dados. A rede IP subjacente garante a qualidade dos serviços implantados sobre ela (além de garantir, entre outras coisas, o serviço de voz), e comunicações seguras. Muitas vezes o serviço VoIP é implementado sobre VPNs, onde os clientes usufruem de um serviço de rede privado que interliga os seus locais remotos através de uma infra-estrutura de rede compartilhada. Neste contexto, os benefícios destas ligações virtuais incluem um maior grau de confiança para o cliente e uma melhor utilização dos recursos para o provedor de serviços.

Para que novos serviços possam ser implementados satisfatoriamente, como é o caso dos serviços de voz e de redes privadas virtuais sobre IP, que são mais complexos, onde é possível termos IP em conjunto com MPLS, é fundamental a forma de gestão destes serviços, incluindo aqui todas as fases do seu ciclo de vida: planificação, provisão, exploração e facturação. Como resultado, a maioria dos fornecedores começa a incluí-los nas suas soluções integrados, com serviços "de fábrica", o que não impede que sejam necessárias algumas adaptações às necessidades específicas de cada operador. A chave para a correcta implantação destes serviços reside na eficácia e fiabilidade dos sistemas de gestão, que devem permitir processos tais como a provisão automática extremo a extremo destes serviços, a gestão centralizada de falhas e uma facturação acertada e oportuna. Assim se reduzirão os *Total Cost of Ownership* (TCO) e se aumentarão os *Return on Investment* (ROI) do fornecedor de serviços.

A convergência de voz e de dados trará consigo novos serviços (por exemplo os serviços, que além de voz e dados, oferecem televisão sobre IP), que coexistirão com outros serviços tais como a videoconferência, aluguer de software, *Video on Demand* (VOD), serviços de formação, entre outros, alguns dos quais estão já disponíveis, embora nem sempre utilizando tecnologia IP.

Estes novos serviços, e muitos dos já existentes, deverão ser fornecidos a pedido, pelo que as novas redes e sistemas que as operam deverão ser capazes de suportar a

provisão e activação automática e imediata dos pedidos dos utilizadores finais. É neste contexto que surgem os portais de cliente [27], perfeitamente integrados com a arquitectura de sistemas OSS de provisão e activação dos operadores e prestadores de serviços, bem como com os sistemas de facturação e gestão de serviços que estas empresas possuem. Este tipo de aplicações permitiria receber pedidos por parte dos utilizadores finais e incluí-los imediatamente na cadeia de provisão de serviços, já que com a rede a prestar automatização de processos extremo a extremo, o utilizador poderia usufruir imediatamente do serviço solicitado.



# **Capítulo 5: Activação automática de Serviços - Caso prático**

O seguinte capítulo apresenta algum do trabalho desenvolvido pelo departamento de Sistemas de Suporte às Operações, da PT Inovação [28], no âmbito de activação de serviços [29]. Com o constante desenvolvimento e proliferação de tecnologias e recursos e de bens provisionáveis, foi necessário definir uma arquitectura de uma Plataforma de Provisão e Cadastro com capacidades supratecnológicas e suprafabricante, de modo a simplificar a adição de novos serviços e tecnologias assim como novas configurações, como forma de responder em tempo útil às necessidades do mercado. Como tal, os operadores que consigam uma melhor aproximação a este cenário, obterão a vantagem competitiva sobre aqueles que a cada novo serviço, não possuam um enquadramento na sua arquitectura que facilite a sua introdução ou sobre aqueles que possuam dependências marcadas com Fabricantes/Tecnologias/Serviços.

## **5.1. Arquitectura de referência de Provisão de Serviços**

Nos dias de hoje, o ambiente onde se inserem as redes de telecomunicações tende a assentar num modelo de liberalização de redes e serviços e na respectiva operação com variados operadores concorrentes ou com acordos entre si a fornecerem serviços como um todo aos clientes. Então, as empresas de telecomunicações têm de assumir na sua estratégia, a pirâmide TMN [30] Isto significa obrigar a que a tecnologia derive da estratégia e das necessidades reais do negócio e que a gestão das redes e serviços tenha a capacidade de resposta eficaz aos desafios que se colocam (ver Figura 15).

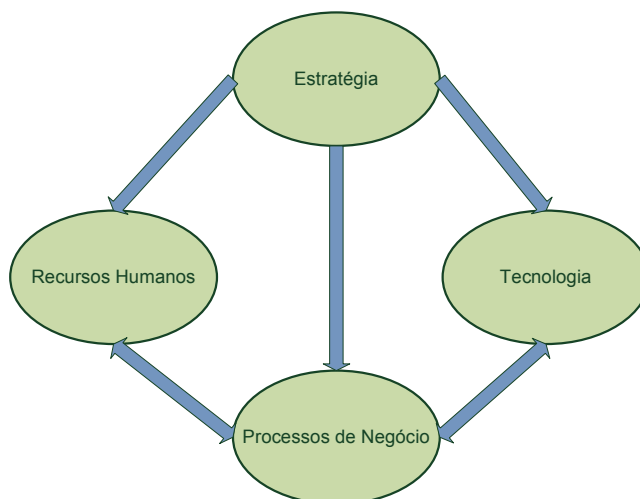


Figura 15 – Estratégia da Empresa.

- **Estratégia:** define as iniciativas necessárias e estabelece as metas a atingir;
- **Recursos Humanos:** facilitar e capacitar a comunicação, a formação, a motivação e a avaliação do desempenho;
- **Tecnologia:** aplicar de forma inovadora a tecnologia nos processos físicos e a tecnologia da Informação;
- **Processos de Negócio:** inventar, reinventar, otimizar processos físicos, informação e decisão/acção.

Torna-se indispensável alinhar os sistemas de informação com os processos de negócio. Aplicar este conceito aos processos de negócio leva à criação de soluções que ligam as estratégias às tecnologias que as suportam, integrando-as com todos os sistemas de informação existentes dentro da empresa. Assim, cada mudança que se produza provoca modificações nas diferentes camadas (*layers*) da estrutura, de forma automática e detectável. Surge a questão de como podem as organizações ajustar a sua visão e a sua estratégia à gestão/execução do negócio. A resposta: desenvolvendo novas infra-estruturas tecnológicas capazes de aliar a estratégia de negócio aos sistemas, conseguindo assim poupanças significativas nos custos de operação.

Uma organização verdadeiramente ágil é aquela que é capaz de responder a cada nova oportunidade aos vários níveis:

- nas muitas camadas da sua arquitectura de negócio;
- no modelo de negócio;
- nos seus processos;
- na arquitectura tecnológica;

de forma a operar no mercado mais depressa do que a sua concorrência. É uma organização que reduz os custos mediante a eliminação de redundâncias nas suas aplicações, nos dados e nas infra-estruturas, que reduz os ciclos de tempo nos seus processos de negócio e que aumenta a sua própria agilidade, capitalizando ao máximo em qualquer evento e reagindo de forma imediata a qualquer mudança.

O Modelo de Referência que se apresenta na Figura 16 resulta da análise do modelo eTOM, do TMForum.

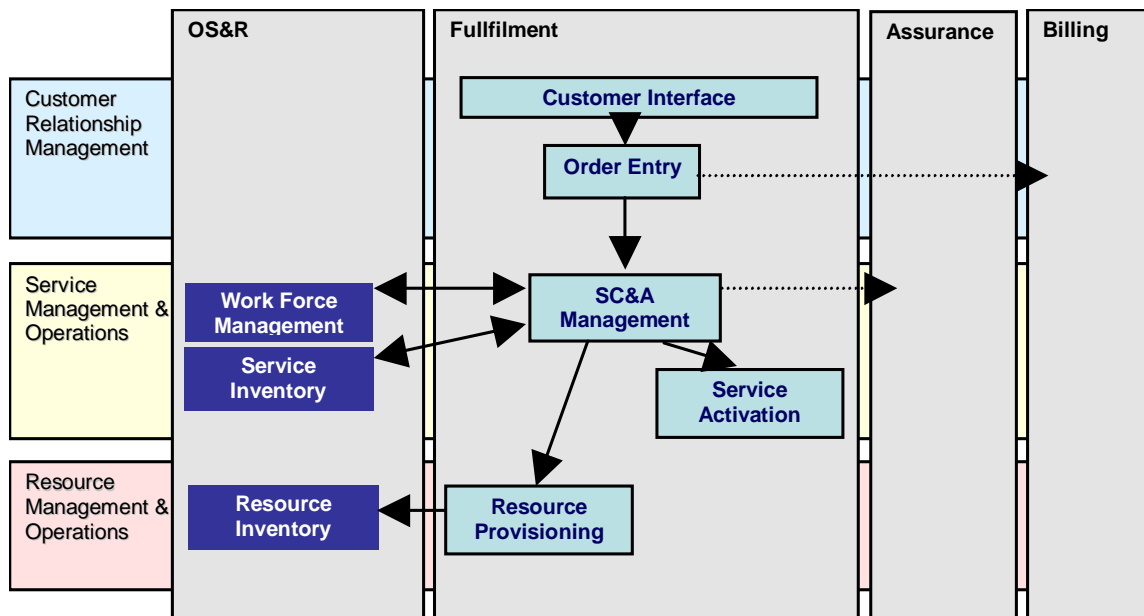


Figura 16 – Modelo de Referência do Processo de Provisão de Serviços [31].

Os processos de *Fulfillment* interagem com os processos de *Billing* e *Assurance* e ainda com uma área genérica de suporte designada por *Operations Support & Readiness*

(OS&R). O grupo de processos que constitui o *OS&R* é responsável pelo suporte dos processos *Fulfillment*, *Assurance* e *Billing (FAB)* e por assegurar a prontidão operacional nas áreas de provisão, facturação e garantia do serviço. De um modo geral, estes processos estão relacionados com actividades que decorrem menos em tempo-real do que as dos processos *FAB*. De acordo com o modelo de referência, enquadram-se nesta área todos os processos de suporte a cadastros, seja de serviço ou de recurso, e ainda de gestão da força de trabalho.

Segue uma breve descrição de cada um dos módulos, segundo [31], com maior ênfase no módulo de *Service Activation*, que se enquadra no âmbito do tema desta tese.

- Os processos relativos ao *Customer Relationship Management (CRM)* têm a responsabilidade de efectuar a interface com o cliente, atendendo os seus pedidos e dando ordem de execução (*Order Entry* → *Order Management*) após determinadas validações (exemplo: existência de crédito do cliente).
- O processo *Service Configuration & Activation Management* tem a responsabilidade de orquestrar as tarefas envolvidas na concretização do pedido, nomeadamente:
  - Notificar o processo de *WFM* para a execução de trabalhos de exterior (exemplo: instalação de equipamento nas instalações do cliente).
  - Carregar a descrição do serviço e criar uma instância do mesmo no inventário de serviço.
  - Interagir com o processo de *Resource Provisioning*, no sentido de efectuar a reserva de recursos no inventário de recursos, para o serviço pedido.
  - Passar a informação necessária ao processo *Service Activation*, de forma a que este proceda à activação do serviço.
  - Actualizar o estado do serviço no *Service Inventory*, notificar os processos de *Assurance* para activarem a monitoria de SLAs, notificar o *Billing* para dar início à tarificação do serviço, e finalmente notificar o CRM, da conclusão do pedido.

- O processo *Resource Provisioning* é o responsável por efectuar a reserva de recursos, necessários ao fornecimento do serviço.
- O processo *Service Activation* tem a responsabilidade de executar os comandos de configuração nos elementos de rede ou nos respectivos *Element Managers*, de forma a concretizar a activação do serviço, nomeadamente:
  - Emitir identificadores para os serviços;
  - Requerer a activação e disponibilidade dos recursos;
  - Despoletar a activação/desactivação de todos os elementos instalados;
  - Suportar os processos de teste *end-to-end* do serviço, assegurando que todos os componentes estão operacionais, e que o serviço está a funcionar nos níveis acordados antes da sua activação para o cliente;
  - Informar os processos responsáveis pela actualização e manutenção dos registos da infra-estrutura do serviço;
  - Disponibilizar detalhes de serviço para efeitos de *Assurance*.

Com base no modelo de referência atrás apresentado foram identificados os sistemas principais que permitem conduzir à automatização *end-to-end* do processo de provisão (*Flowthrough Provisioning*) (Figura 17).

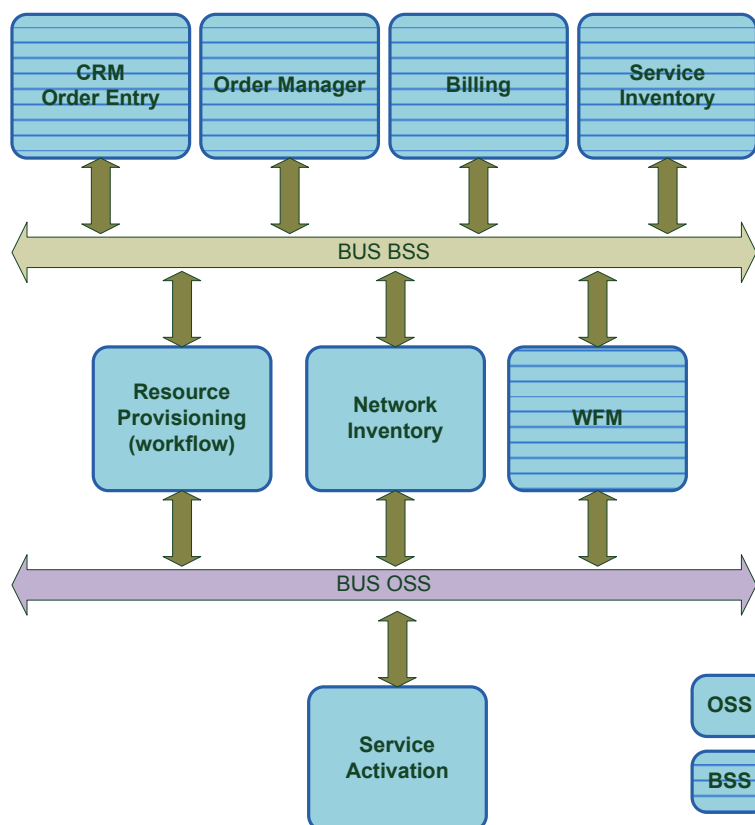


Figura 17 – Arquitectura de Referência. [29]

A arquitectura de referência consiste na identificação do conjunto de sistemas de informação necessários à implementação dos processos de negócio, e na definição da arquitectura de interligação dos mesmos. A arquitectura de referência apresentada na Figura 17 baseia-se nos princípios da arquitectura NGOSS, de interligação de sistemas de BSS/OSS, definida no TMForum. Esta pressupõe a existência de um modelo de informação partilhado por todos os sistemas (SID) e de *buses* integradores que garante o fluxo de mensagens entre sistemas.

Esta arquitectura caracteriza-se pelos seguintes aspectos (ver Figura 18):

- Implementação/Utilização de Interfaces abertas para fácil integração, eliminando a utilização de modelos de informação, protocolos e tecnologias proprietários, i.e. específicos de um dado fornecedor, em favor dos seus correspondentes normalizados;
- Utilização/Implementação de um Bus EAI para suportar a comunicação entre os BSS e os OSS;

- Na integração dos vários OSS poderá ser equacionada a utilização de um Bus de tecnologia mais leve (ex.: utilizando APIs *open source*);
- Um *Order Management* genérico com capacidade multi-serviços que controle todas as actividades necessárias à implementação do desenho do serviço;
- Um cadastro de serviços individualizado, separado do cadastro de recursos;
- Um sistema de *Work Force Management* para suportar as actividades manuais que exijam a gestão de recursos humanos/equipas;
- Um *Service Activation* que disponha de *plug-ins* por tecnologia que permitam a interacção com os vários sistemas de gestão de rede/elemento de rede para proceder à activação *end-to-end* do serviço pedido.

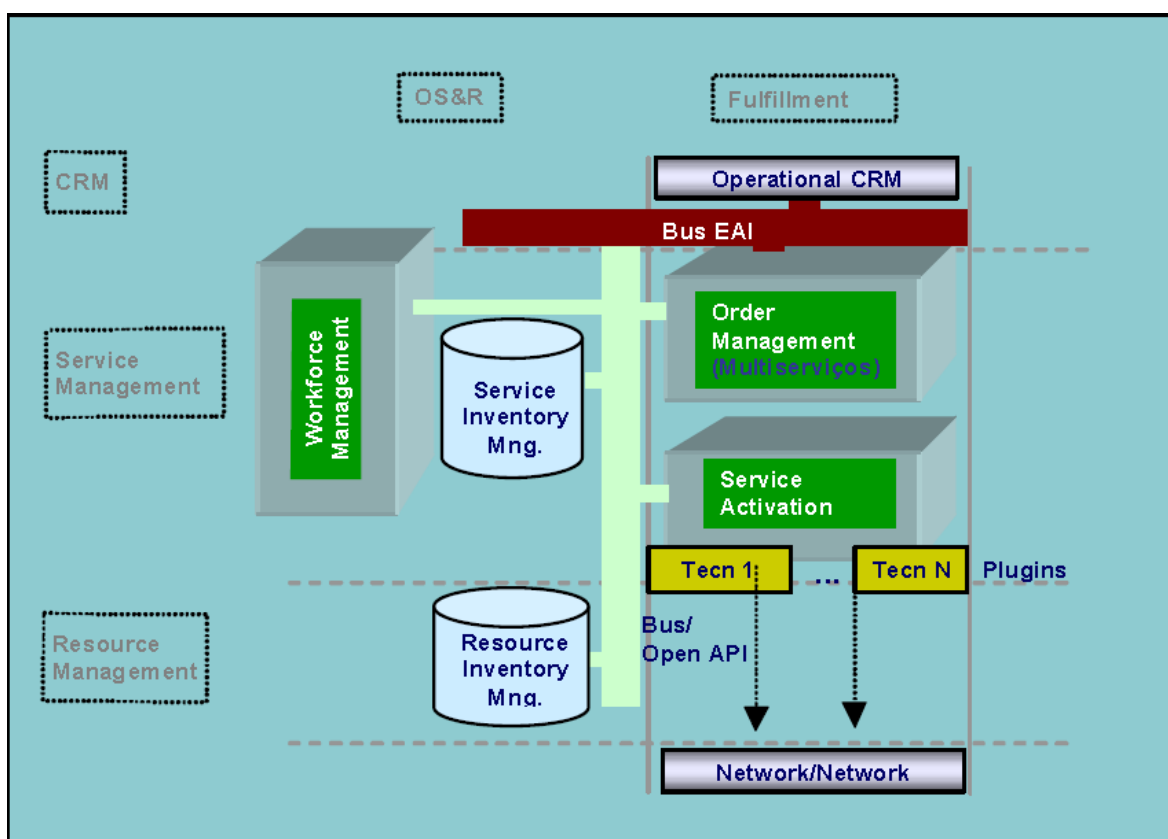


Figura 18 – Provisão de Serviços – Arquitectura de Referência [29].

## 5.2. Network Activator

O caso prático presente nesta tese tem como objectivo especificar e implementar o protótipo de uma plataforma de activação de serviços/recursos, genérica, configurável e multi-serviços, de forma a permitir efectuar a provisão automática e a reconfiguração dinâmica de recursos. É identificado com o nome *Network Activator*, ou NA, e em termos de posicionamento na arquitectura de referência, o caso prático aqui apresentado pretende ser uma plataforma de *Service Activation* Multi-Serviços, constituindo uma peça relevante na automação da provisão de serviços *end-to-end*, vulgarmente designada de *Flow Through Provisioning* (FTP). Estes processos automáticos de provisão (FTP ou *Zero Touch Flow*) são muito difíceis de alcançar, em toda a linha de negócio (serviços). Por exemplo, quando se trata da decomposição de uma ordem de serviço solicitada para serviços sobre a rede (seja ela PSTN ou banda larga), alguns engenheiros de campo são obrigados a estar envolvidos no processo e, portanto, 100% da automação não é possível. Os processos automáticos de provisão funcionam muito bem para serviços ADSL, por exemplo, a partir do momento que um cliente compra o equipamento, instala-o em sua casa e faz uma chamada telefónica para a sua activação, até ao momento em que o serviço é disponibilizado e o cliente recebe a primeira factura. No entanto, para serviços de nova geração, os *Zero Touch / Flow Through Provisioning* não são uma opção mas sim uma exigência do dia-a-dia.

## 5.3. Definição de Requisitos

Os requisitos de um sistema não passam de descrições de serviços providenciados pelo sistema assim como restrições operacionais a ele associados. Estes requisitos reflectem a necessidade que os utilizadores do sistema possuem para resolver problemas como por exemplo o modo de operação do dispositivo, como fazer um pedido, procurar informação, etc. Todo o processo de recolha de análise destes serviços e restrições tem o nome de Engenharia de Requisitos [32].

O contexto de serviço utilizado ao longo dos requisitos refere-se ao conceito de *Resource Facing Service* (RFS), definido no modelo SID [19] (Figura 19 e Figura 20), do TMForum, que apresenta o serviço na perspectiva dos recursos lógicos que o



implementam, distinguindo-o do serviço do ponto de vista do cliente. Neste modelo um serviço (RFS) é composto por um conjunto de *Logical Resources* (LR).

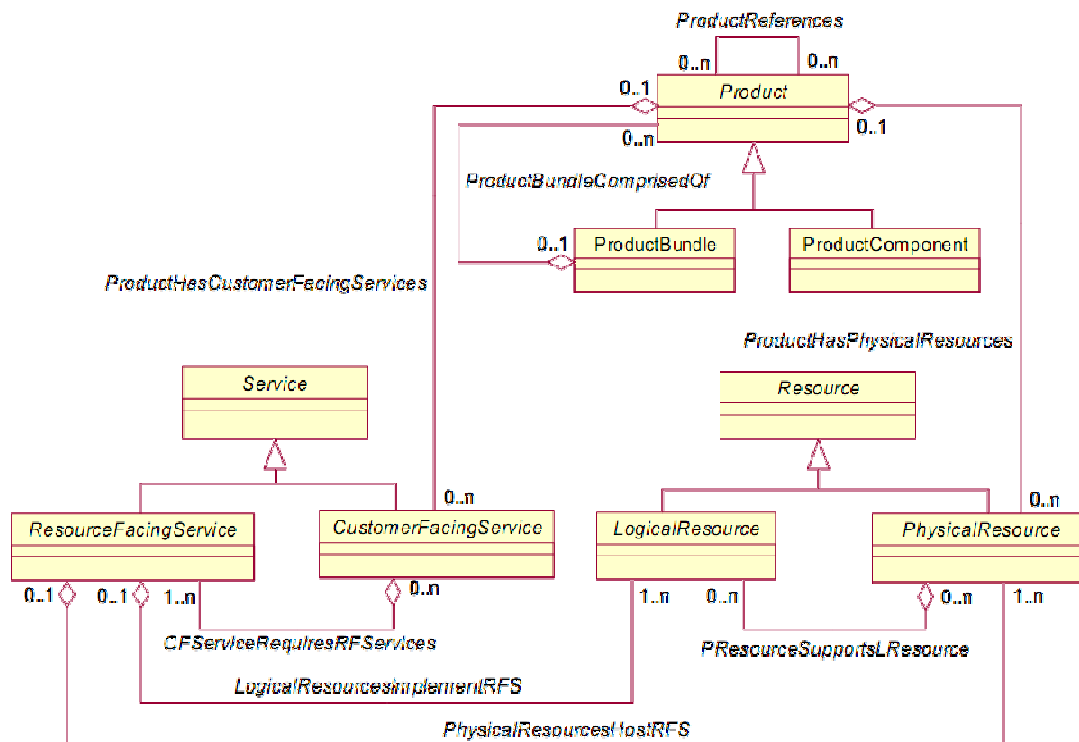


Figura 19 – Modelo SID - Relação entre Produto, Serviço e Recurso.

A Figura 19 mostra a distinção entre serviços que são visíveis pelo cliente, designados por *Customer Facing Services* (CFS), que fazem parte de um produto comercial, e serviços que suportam a sua implementação, denominados *Resource Facing Services* (RFS), que não têm visibilidade para o cliente.

A figura 20 ilustra o conceito de *Service Specification*, que define um serviço. A relação *RFSpecHasResourceSpecs* mostra que a especificação de um RFS é composta pelas especificações dos LR que implementam o serviço. Este conceito foi utilizado na entidade *Service Descriptor*, do Network Activator. O objectivo é construir um catálogo de serviços (RFS) com base num catálogo de recursos lógicos (LRs). A especificação dos RFS e dos LR será baseada em *schemas XML*.

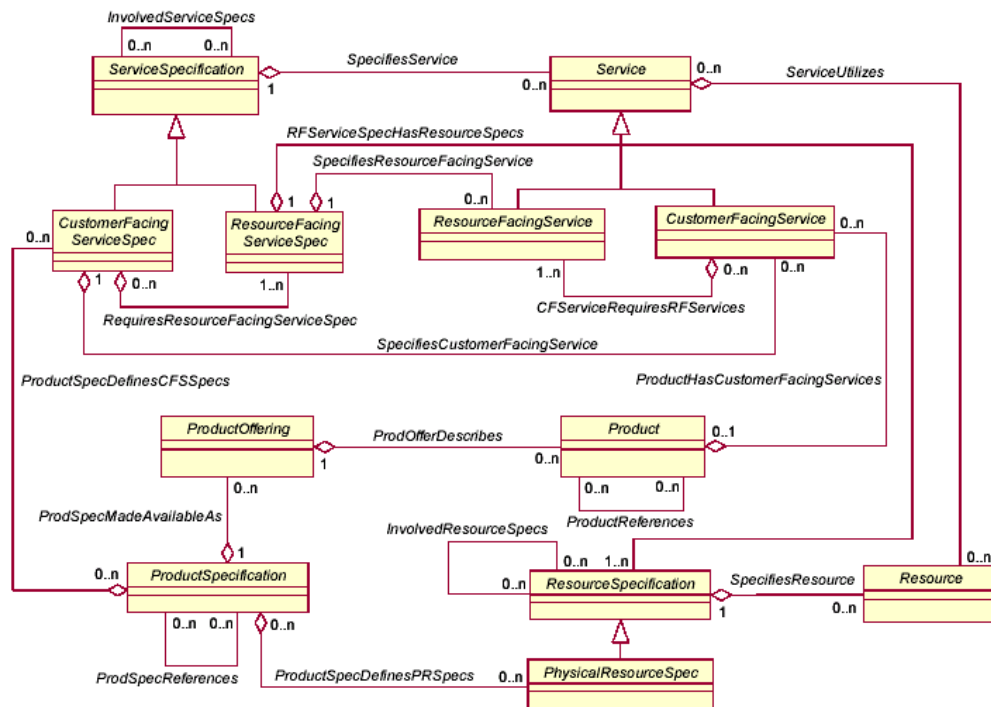


Figura 20 – Conceito *Service Specification*.

### 5.3.1. Requisitos Funcionais

Requisitos funcionais descrevem o comportamento do sistema a implementar. Estes requisitos dependem do tipo de software a desenvolver, os serviços que este fornece, assim como o modo de reacção ou comportamento a particulares interacções com o utilizador. Em alguns casos, os requisitos funcionais poderão explicitar o modo como o sistema não se deve comportar.

Os requisitos funcionais do NA foram agrupados da seguinte forma:

- Requisitos Genéricos (RF.GNR.xx)
- Requisitos Externos (RF.EXT.xx)
- Requisitos de Arquitectura (RF.ARQ.xx)
- Requisitos de Interfaces Externas (RF.IF.xx)

## Requisitos Genéricos

Código	Descrição	Prioridade
RF.GNR.01	O NA deve enquadrar-se no processo de provisão de serviços, implementando a funcionalidade de activação de serviços/recursos.	ELEVADA
RF.GNR.02	O NA deve possibilitar a provisão automática e (re)configuração dinâmica de recursos em tempo real.	ELEVADA

Tabela 1 – Requisitos Genéricos

## Requisitos Externos

Código	Descrição	Prioridade
RF.EXT.01	Deve existir um modelo de informação comum, partilhado pelos diferentes sistemas, que descreva os serviços, os recursos e as relações entre ambos.	ELEVADA
RF.EXT.02	Deve existir um repositório centralizado de templates (descritores de serviço) com a descrição do serviço.	BAIXA
RF.EXT.03	Devem poder ser suportadas diferentes versões de especificação do mesmo serviço.	ELEVADA

Tabela 2 – Requisitos Externos

## Requisitos da Arquitectura

Código	Descrição	Prioridade
RF.ARQ.01	Cada novo serviço deve poder ser carregado em <i>runtime</i> no NA. Este deve publicitá-lo na interface <i>northbound</i> .	ELEVADA
RF.ARQ.02	O NA deve ter embebido um <i>microworkflow</i> que controla as operações em vários recursos envolvidos num pedido de serviço.	ELEVADA
RF.ARQ.03	O NA deve implementar um modelo de estados do pedido.	ELEVADA
RF.ARQ.04	O NA deve implementar persistência dos pedidos.	ELEVADA
RF.ARQ.05	O NA deve permitir a instalação de <i>plug-ins</i> (adaptadores) específicos por tipo de recurso a configurar na interface <i>southbound</i> .	ELEVADA
RF.ARQ.06	Os <i>plug-ins</i> devem mapear operações genéricas, recebidas da interface <i>northbound</i> , em operações específicas e dependentes da interface com o recurso.	ELEVADA
RF.ARQ.07	Os <i>plug-ins</i> devem poder ser instalados em <i>runtime</i> .	ELEVADA
RF.ARQ.08	O NA deve conter <i>plug-ins</i> genéricos para interfaces <i>southbound</i> CLI, SNMP, Shell-Script, http e FTP. Cada instância destes <i>plug-ins</i> deve ser configurável, para implementar o comportamento do recurso gerido.	ELEVADA
RF.ARQ.09	Um <i>plug-in</i> deve poder suportar diferentes versões do mesmo recurso.	ELEVADA
RF.ARQ.10	Deve ser possível existir um <i>plug-in</i> por versão do recurso gerido.	ELEVADA
RF.ARQ.11	O NA deve suportar <i>plug-ins</i> para interfaces <i>southbound</i> Radius, Soap/webServices, Corba, Database, Flat-File.	BAIXA

Código	Descrição	Prioridade
RF.ARQ.12	Deve ser possível enviar toda a informação ( <i>bundle</i> ) de vários serviços, incluídos no mesmo pedido, para o mesmo <i>plug-in</i> .	ELEVADA
RF.ARQ.13	Deve ser possível programar a sequência de execução de acções em recursos de um mesmo serviço.	ELEVADA
RF.ARQ.14	Deve ser possível programar regras de selecção de <i>plug-ins</i> baseadas em atributos dos recursos geridos.	ELEVADA

Tabela 3 – Requisitos da Arquitectura

### Requisitos de Interfaces Externas

Estes requisitos identificam as funcionalidades que devem ser disponibilizadas pelo NA, nas interfaces *northbound* e/ou *southbound*. A sigla IFN identifica a interface *northbound* e a sigla IFS identifica a interface *southbound*.

Código	Descrição	Prioridade
RF.IFN.01	O NA deve disponibilizar uma interface <i>northbound</i> , genérica, para interacção com plataformas de <i>Order Management</i> , GW B2B, e portais.	ELEVADA
RF.IFN.02	O NA deve permitir operações síncronas e assíncronas.	ELEVADA
RF.IFN.03	O NA deve aceitar pedidos de activação, alteração e desactivação de serviços.	ELEVADA
RF.IFN.04	O NA deve processar vários pedidos numa única operação. ( <i>mass/bulk provisioning</i> ).	MÉDIA
RF.IFN.05	O NA deve permitir o escalonamento de pedidos.	BAIXA
RF.IFN.06	O NA deve permitir operações de consulta sobre o estado dos pedidos.	MÉDIA
RF.IFS.01	O NA deve suportar interfaces <i>southbound</i> CLI, SNMP, <i>Shellscript</i> e <i>Http</i>	ELEVADA
RF.IFS.02	O NA deve suportar interfaces <i>southbound</i> Radius, Soap/webServices, Corba, Database, Flat-File.	BAIXA
RF.IFS.03	Devem ser suportadas interfaces <i>southbound</i> assíncronas	ELEVADA

Tabela 4 – Requisitos de Interfaces Externas

### 5.3.2. Requisitos Não Funcionais

Requisitos não funcionais, como o próprio nome indica, são requisitos que não estão directamente relacionados com funções específicas do sistema. Podem definir restrições nos serviços ou funções oferecidas pelo sistema, tais como restrições de tempo, restrições de interface, restrições de desempenho, etc. Poderão ainda definir restrições de entrada e

saída de dados do dispositivo, assim como a representação de dados nas interfaces de sistema.

Código	Descrição	Prioridade
RnF.01	A tecnologia a utilizar na interface <i>northbound</i> deve ser preferencialmente <i>Web Services</i>	ELEVADA
RnF.02	Os serviços devem ser descritos com base em <i>templates XML</i>	ELEVADA
RnF.03	A configuração de comportamento de recursos, que utilizam <i>plug-ins</i> genéricos CLI, SNMP, Shellscrip e Http, deve ser descrita em XML.	ELEVADA
RnF.04	A arquitectura do NA deve garantir escalabilidade, redundância e independência de sistema operativo.	ELEVADA

Tabela 5 – Requisitos Não Funcionais

## 5.4. Arquitectura do Sistema

O NA foi desenhada com o objectivo de permitir a implementação de um sistema flexível, e configurável, e em que o suporte para um novo serviço ou recurso (*plug-in*), se resume à construção de *scripts XML*, sem necessidade de desenvolvimento de *software* aplicacional.

Os requisitos identificados anteriormente, serviram de guia para as fases de análise e desenho, tendo-se chegado à arquitectura genérica representada na Figura 21, que mostra uma perspectiva global e modular das funcionalidades do sistema.

O desenho do sistema segue os seguintes pressupostos:

- Existência de um template (*schema XML*), que define o modelo genérico de um serviço.
- Capacidade de criação de descritores de serviço (*schemas XML*), com base no template genérico.
- Conhecimento partilhado dos descritores de serviço pelos sistemas de *Order Management* e pelo(s) *Network Activator(s)*.

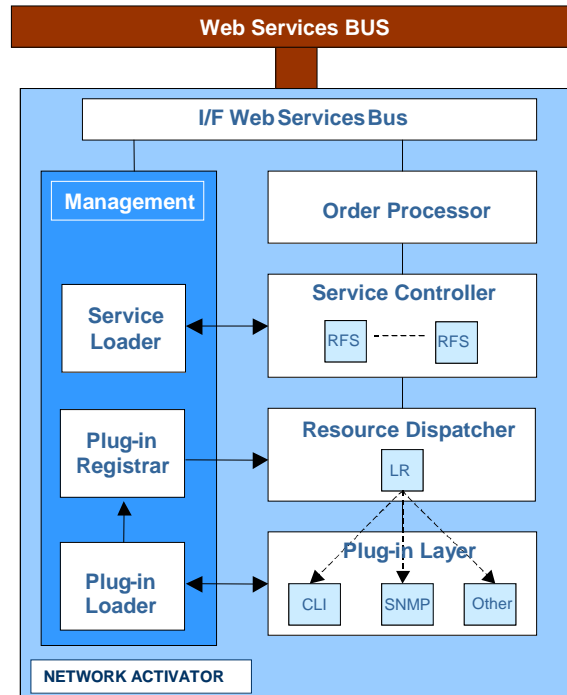


Figura 21 – Arquitectura do Sistema.

O NA foi idealizado para ser uma arquitectura modular, organizada por camadas, numa perspectiva *top-down*, seguindo o fluxo: pedido → serviço → recurso → elemento de rede, composta pelos seguintes módulos:

- *Web Services Interface*: Interface genérica, independente do serviço, que disponibiliza facilidades de invocação de pedidos de activação, alteração e cancelamento de serviços. Nas operações de pedido deve ser passado um documento XML, com a informação dos serviços.
- *Order Processor*: Processa os pedidos provenientes da interface *northbound*. Suporta pedidos síncronos (tempo real) e assíncronos (*bulk provisioning*). Valida se os dados do pedido estão de acordo com o template genérico (*schema XML*) de definição de serviços e extrai os serviços individualmente. Passa o XML que caracteriza os serviços para o módulo *Service Controller*.
- *Service Controller*: Valida se os serviços recebidos do *Order Processor*, são suportados pelo *Network Activator*, e decompõe-os em recursos lógicos. Controla a execução de cada pedido nos vários recursos, através de *microworkflow* interno. Passa o XML que caracteriza os recursos para o módulo *Resource Dispatcher*.
- *Resource Dispatcher*: Identifica o contexto dos recursos lógicos, recebidos do *Service Controller*, e procura no módulo *Plug-in Registrar*, os *plug-ins*

responsáveis pelo controlo dos recursos. Agrega os recursos por *plug-in* e passa o XML que caracteriza os recursos para os respectivos *plug-ins*.

- *Plug-in*: Identifica o contexto dos recursos lógicos, recebidos do módulo *Resource Dispatcher*, e procura a *script* que lhe permita transformar o XML recebido em comandos para o NE/EMS. No caso de *plug-ins* específicos esta transformação está embebida em código. De notar que, o modo de processamento de um *bundle* de recursos lógicos é da inteira responsabilidade do *plug-in*, dependendo da sua configuração.

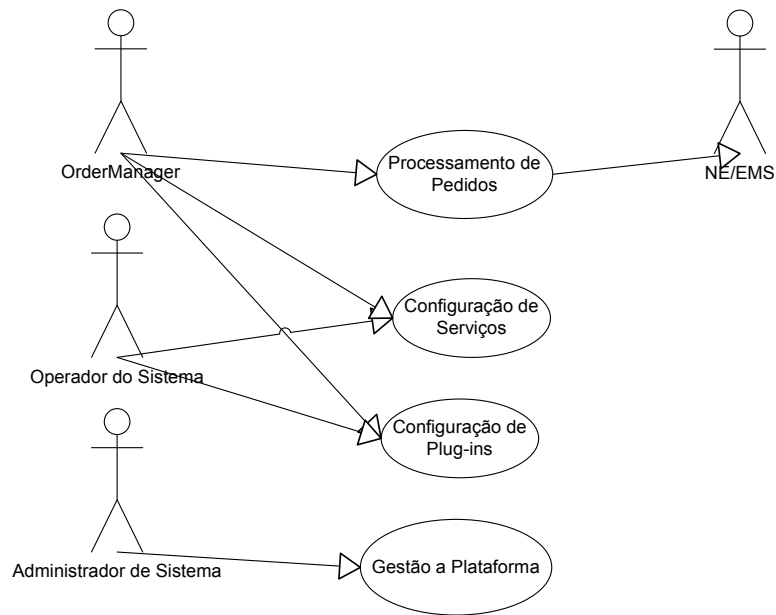
O *Network Activator* possui também um módulo de gestão responsável pela configuração de serviços e *plug-ins* no sistema. Este módulo permite:

- Carregamento de descritores serviço no *Service Controller*, para que este módulo tenha conhecimento dos serviços suportados.
- Carregamento e registo de *plug-ins*, para que o *Resource Dispatcher* possa determinar o *plug-in* a utilizar no contexto do recurso a configurar.

## 5.5. Perspectiva Funcional

A perspectiva funcional apresentada identifica os actores e o modelo de *Use Cases* do sistema (ver Figura 22). O comportamento do sistema para com o utilizador e vice-versa é capturado através da análise de Casos de Utilização chamados *Use Case* (Casos de Utilização). Um Caso de Utilização é a especificação de uma sequência de interações entre um sistema e os agentes externos que utilizam esse sistema e representa quem faz o quê (interage) com o sistema, sem considerar o comportamento interno do mesmo.

A identificação e modelação da funcionalidade do sistema em Casos de Utilização não é uma “ciência exacta”. Dois analistas podem chegar a modelos diferentes e, mesmo assim, correctos. Os modelos iniciais devem ser simples, cobrir a funcionalidade pretendida e ser úteis para dialogar com os utilizadores finais.



**Figura 22 – Perspectiva Funcional do Sistema.**

## Actores

### **Actores Humanos:**

- Administrador do Sistema
  - Responsável pelas funções de administração da plataforma computacional (gestão de base de dados, limpeza de *logs*, etc).
- Operador do Sistema
  - Pode executar localmente (via linha de comandos) todas as operações disponíveis, via interface *Web Services*, nomeadamente: carregamento de serviços, carregamento de *plug-ins*, etc.

### **Actores Sistemas:**

- Order Manager
  - Representa genericamente os sistemas de provisão de serviços, gateways B2B e Portais de Serviços.
  - Emitem pedidos de activação, alteração e cancelamento de serviços.
  - Podem consultar no *Network Activator*, os serviços disponíveis e os tipos de recursos que podem configurados.

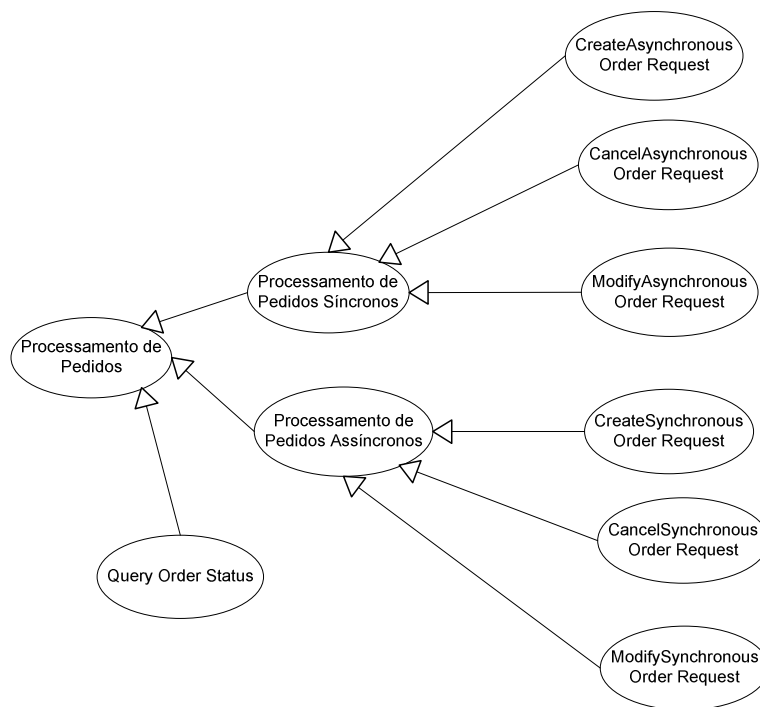


- NE/EMS
  - Representa as entidades que recebem pedidos do *Network Activator*, para executarem os comandos de activação dos serviços nos recursos de rede. Podem ser os próprios elementos de rede (NE) ou os respectivos *Element Managers* (EMS), quando não é permitido o acesso directo aos NEs.

### Use Cases

Estes foram os *Use Cases* de alto nível identificados:

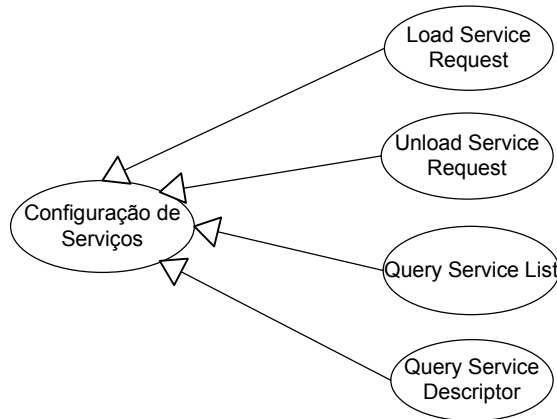
- Processamento de Pedidos (Figura 23);



**Figura 23 – Use Cases de Processamento de Pedidos.**

Estas *Use Cases* são utilizadas para invocar: pedidos de activação (Create...), de alteração (Modify...) e de cancelamento (Cancel...) de serviços; e operações de consulta do estado dos pedidos (QueryOrderStatus). Os pedidos podem requerer um processamento síncrono, para operações com requisitos de tempo real, ou assíncrono, para *mass/bulk provisioning*. No primeiro caso o cliente fica à espera do resultado do pedido, enquanto no segundo caso, o resultado do pedido é enviado posteriormente para o cliente.

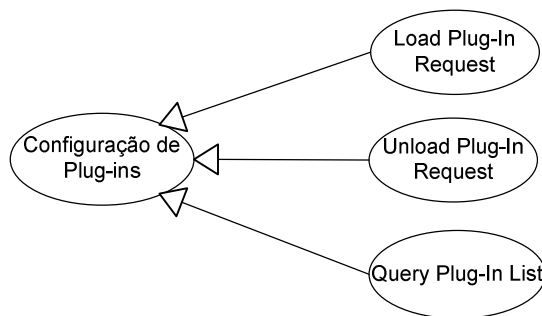
- Configuração de Serviços (Figura 24);



**Figura 24 – Use Cases de Configuração de Serviços.**

Os *Use Cases* de Configuração de Serviços são utilizados pelo actor *Operador do Sistema*, principalmente para carregar (load) e remover (unload) descritores de serviços no *Network Activator*. Existem ainda dois *Use Cases* que permitem consultar a lista de serviços disponíveis no *Network Activator* (*QueryServiceList*) e pedir o descriptor de determinado serviço (*QueryServiceDescriptor*).

- Configuração de *Plug-ins* (Figura 25);



**Figura 25 – Use Cases de Configuração de *Plug-ins*.**

Os *Use Cases* de Configuração de *Plug-ins* são utilizadas pelo actor *Operador do Sistema*, principalmente para carregar (*load*) e remover (*unload*) *plug-ins* no *Network Activator*. Adicionalmente a *Use Case QueryPlug-inList* permite consultar a lista de *plug-ins* por tipos de recurso, que existem carregados no *Network Activator*.

- Gestão da Plataforma;

## 5.6. Perspectiva Lógica

A perspectiva lógica identifica os módulos constituintes do sistema, as interfaces entre eles, e o modelo de informação (entidades envolvidas). No capítulo 5.4, foi feita uma breve apresentação da arquitectura do *Network Activator*, tendo sido identificados os principais módulos. Aqui, pretende-se aprofundar a visão global aí fornecida e pormenorizar o comportamento de cada módulo. A figura 26 mostra a estrutura de módulos, as entidades e as interfaces do *Network Activator*.

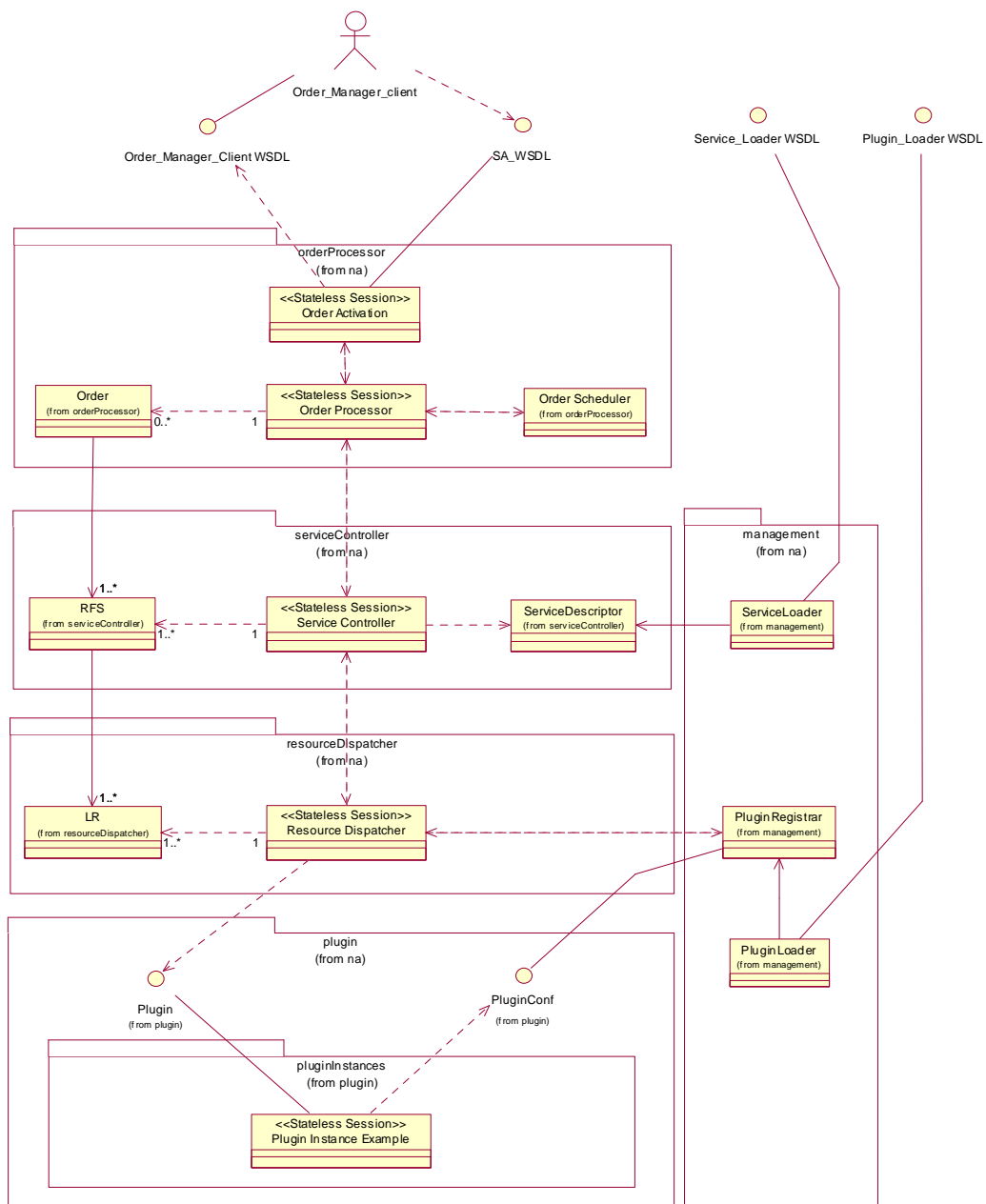


Figura 26 – Estrutura de módulos do *Network Activator*.

Na Figura 27 estão apresentadas as entidades mais relevantes do sistema:

Order	RFS	LR	ServiceDescriptor
-clientID	-rfsID	-lrID	-descriptorID
-orderID	-rfsType	-lrType	-descriptorType
-priority	-rfsVersion	-lrVersion	-descriptorVersion
-syncType	-rfsInfo	-lrManagementMethod	-descriptorInfo
-completionDate		-lrInfo	
-orderInfo			

Figura 27 – Entidades Relevantes do Sistema.

- A entidade *Order* é passada nas operações (*createOrder*, *modifyOrder* e *cancelOrder*), disponíveis na interface *northbound*. De salientar que esta entidade é totalmente independente do serviço.
- A entidade RFS (*Resource Facing Service*) identifica o serviço que se pretende activar. Contém encapsulada no atributo *rfs Info* a informação dos recursos (LRs) envolvidos no serviço e atributos de serviço comuns a todos os recursos.
- A entidade LR (*Logical Resource*) é composta por uma lista de pares atributo/valor (atributo *lr Info*), que caracterizam determinado tipo de recurso.
- A entidade *Service Descriptor* descreve um tipo/versão de serviço RFS, identificando o tipo de recursos lógicos que o constituem.

A tabela seguinte pormenoriza a descrição destas entidades baseando-se no modelo SID, do *TMForum*.

Entidade	Atributo	Descrição
<b>Order</b>	client ID	Identifica o cliente que emitiu a ordem
	order ID	Identifica univocamente a ordem no contexto do cliente
	priority	Prioridade da ordem
	sync Type	Tipo de serviço: síncrono ou assíncrono
	completion Date	Data/hora em que a ordem deve ser iniciada. Usado apenas para o serviço assíncrono, quando se pretende utilizar a facilidade de <i>scheduling</i>
	order Info	Sequência de RFS que integram a ordem ( <i>string XML</i> )
<b>RFS</b> ( <i>Resource Facing Service</i> )	rfs ID	Identifica univocamente o RFS dentro de uma ordem
	rfs Type	Identifica o tipo de serviço

	rfs Version	Identifica a versão de especificação do serviço
	rfs Info	Sequência de entidades LR, que implementam o serviço ( <i>string XML</i> ). Inclui ainda atributos comuns a todos os LRs.
<b>LR</b> ( <i>Logical Resource</i> )	lr ID	Identifica univocamente a entidade LR, num RFS
	lr Type	Identifica o tipo de LR
	lr Version	Identifica a versão de especificação do LR
	lr ManagementMethod	Identifica o protocolo de gestão de acesso ao recurso. Usado para determinar qual o plug-in responsável pelo controlo do recurso.
	lr Info	Atributos variáveis dependendo do tipo/versão do LR ( <i>string XML</i> )
<b>Service Descriptor</b>	descriptor ID	Identifica o descritor que especifica o serviço, na lista de descritores
	descriptor Type	Identifica o tipo de descriptor
	descriptor Version	Identifica a versão do descriptor
	descriptor Info	Especifica os tipos de LRs que compõem um RFS, especificado por este descriptor.

Tabela 6 – Descrição das entidades relevantes do Sistema.

### 5.6.1. Descrição dos Módulos

São apresentadas de seguida a descrição dos módulos existentes no sistema.

#### Order Processor

O módulo *Order Processor* é responsável pelo processamento de ordens recebidas via interface *northbound*, gerindo os pedidos veiculados nesta interface, controla a sua execução e devolve o respectivo resultado. Esta interface disponibiliza dois tipos de serviço: serviço síncrono, para pedidos de tempo real; e serviço assíncrono para operações de *mass/bulk provisioning*.

A arquitectura lógica do módulo está estruturada em 3 níveis:

**Nível 1:** nível de interacção com o interface northbound (Web Service),

- Implementado pelo componente *Order Activation*. Este componente é responsável pela recepção dos pedidos, pela validação dos parâmetros dos respectivos pedidos,

pela delegação dos pedidos para a lógica de negócio exposto por um nível de processamento e pelo envio de respostas ao cliente.

- A validação dos pedidos neste nível assegura que todos os pedidos passados para o nível de negócio (processamento) são válidos, evitando assim estimular o nível de negócio com pedidos inválidos.
- No caso de recepção de pedidos com documentos XML, o *Order Activation* deverá validar o respectivo *XML schema*.
- As respostas a enviar ao cliente poderão ser ficheiros XML, passados como objectos do tipo *String*.

**Nível 2:** nível de processamento (ou de lógica de negócio),

- Implementado pelo componente *Order Processor*. Este componente mapeia os parâmetros do interface para objectos específicos do domínio do problema (por exemplo, a Ordem). O agendamento das Ordens é gerido pelo componente de *OrderScheduler*, para o caso de pedidos assíncronos.
- A transição de estados da Ordem é gerida por este componente, bem como a respectiva activação de serviços e agregação dos respectivos resultados.

**Nível 3:** nível de interacção com o módulo Service Controller,

- De forma a dar seguimento à concretização da ordem, o Order Processor invoca o método respectivo na interface com o Service Controller, onde passa a *string XML* da estrutura *OrderInfo*, que caracteriza os serviços RFS envolvidos na ordem. O *Service Controller*, devolve RFS a RFS, o resultado de sucesso/insucesso da operação no RFS, numa *string XML*. É da responsabilidade do Order Processor agregar o resultado de todos os RFS de uma ordem.

A Ordem é uma entidade gerida pelo módulo *Order Processor*, possuindo um conjunto de estados:

- **CREATED** - estado inicial da Ordem, apenas para o caso de pedidos assíncronos;
- **RUNNING** - a Ordem transita para este estado após ser dado o início à sua execução, ou no caso de pedidos síncronos, corresponde ao estado inicial;

- COMPLETED - estado final da Ordem, ou seja, a Ordem foi devidamente executada;
- ABORTED - a ordem foi abortada, devido a problemas internos de processamento;
- REMOVED - a Ordem foi removida por indicação explícita do cliente;

Ao fim de um tempo configurável a Ordem será apagada do módulo (END\_STATE). De referir que todas as transições para os estados RUNNING, ABORTED e COMPLETED são controlados pela implementação.

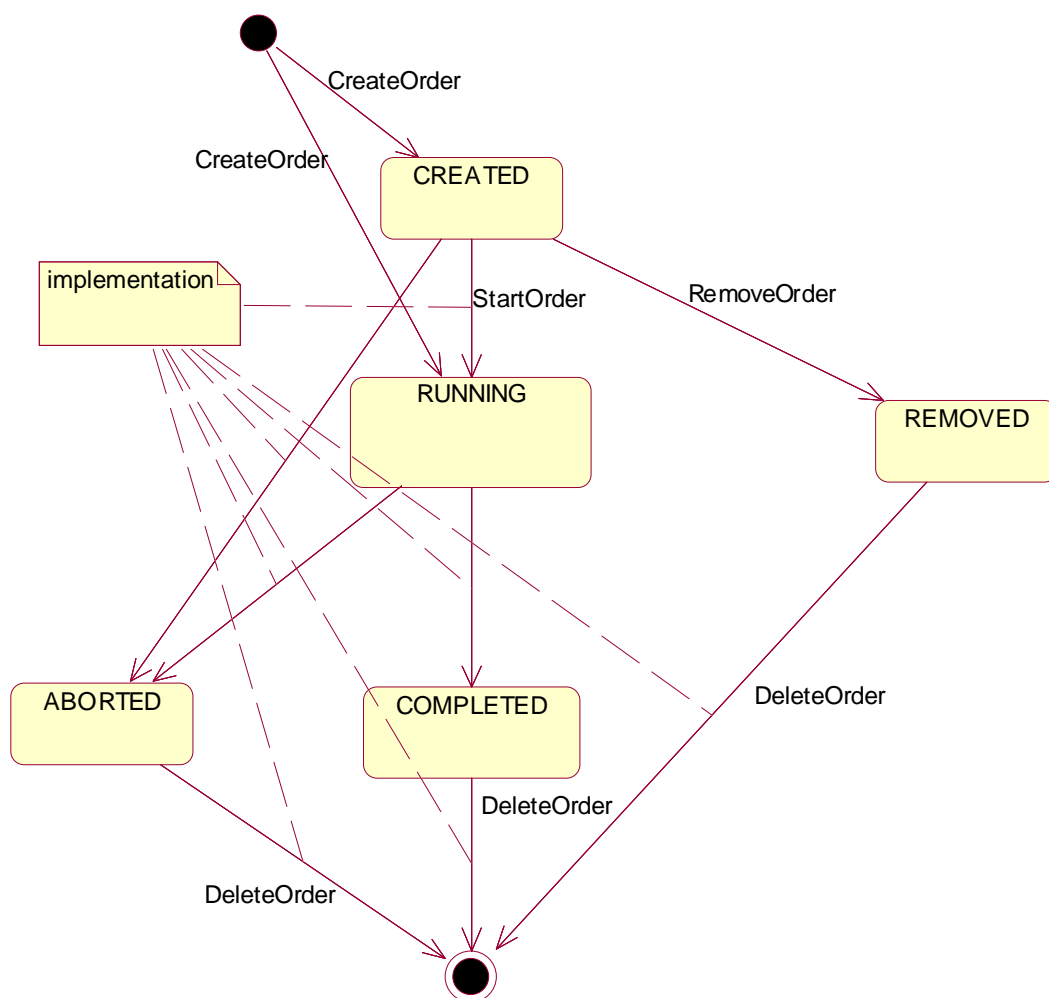


Figura 28 – Máquina de estados da entidade Ordem.

### **Service Controller**

O *Service Controller* é o módulo que controla o processo de activação dos serviços. Disponibiliza as operações de *activateService (XMLOrderInfo)*, *modifyService (XMLOrderInfo)* e *cancelService (XMLOrderInfo)*, nas interfaces síncrona e assíncrona, com o *Order Processor*. Nestas operações, recebe uma *string XML (XMLOrderInfo)*, que contém o grupo de RFS a processar. De seguida valida cada RFS, verificando se está de acordo com o *Service Descriptor* correspondente.

Após esta fase, passa-se à decomposição dos RFS. Cada RFS é decomposto em atributos de serviço (comuns a todos os LRs) e em LRs. Os atributos de serviço são de seguida adicionados a todos os LRs, do RFS.

Por fim, todos os LRs, do RFS recebidos na mesma ordem, são passados *em bundle* para o *Resource Dispatcher*, numa *string XML*.

O *Service Controller* após ter recebido o resultado da operação invocada no *Resource Dispatcher*, devolve ao *Order Processor* o resultado de sucesso/insucesso na activação do serviço. A operação no RFS só é declarada bem sucedida se foram terminadas com sucesso as operações em todos os LRs. Em caso de falha em pelo menos um LR, considera-se que a operação ao nível do RFS falhou. Na resposta para o *Order Processor* é enviada a lista dos LRs, indicando se a operação teve sucesso ou insucesso.

### **Resource Dispatcher**

Este módulo é responsável por encaminhar as entidades LR para o plug-in responsável pela interacção de gestão com os recursos. Disponibiliza as operações de *activateResource (XMLLRs)*, *modifyResource (XMLLRs)* e *cancelResource (XMLLRs)*, na interface com o *Service Controller*. Nestas operações, recebe uma *string XML* que caracteriza os LRs, e com base no atributo *lr ManagementMethod*, caso esteja preenchido, ou na aplicação de regras sobre os atributos do LR, procura no *Plug-in Registrar* o Plug-in que irá interagir com o recurso. Após esta fase de procura de *plug-ins*, o *Resource Dispatcher* agrupa todos os LRs por plug-in e invoca na interface com o Plug-In o método corresponde à acção em curso, onde é passada a *string XML* que agrupa os LRs.



Após a resposta do *Plug-in*, o *Resource Dispatcher* devolve ao *Service Controller* o resultado de sucesso/insucesso da operação efectuada nos LRs. A cada LR pode ser adicionada uma *string XML*, com o *Log* da operação.

### **Plug-in**

O módulo *Plug-in* é um conceito abstracto que pretende representar a agregação dos diferentes *plug-ins* que existem no sistema. Um *plug-in* é responsável pela comunicação de gestão com os recursos (NE/EMS), que efectivamente concretizam o serviço. Detém o conhecimento que lhe permite transformar a informação recebida nas entidades LR, numa sequência de comandos particulares a enviar para um recurso.

## **5.7. Perspectiva Física**

De seguida é apresentado o sistema de um ponto de vista físico, com as descrições dos ambientes de *runtime* do *Network Activator*, identificando requisitos de plataforma computacional, *middleware* e *software* necessários para o funcionamento do sistema.

### **XML**

XML é uma linguagem padrão para a construção de documentos electrónicos com formato simples, textual, estruturado, flexível a mudanças e portátil em diversas plataformas tecnológicas, que tem como objectivo principal descrever informações, tornando-se, assim, ideal para comunicação entre redes heterogéneas. As características do XML permitem que num mesmo documento se apresentem dados e suas respectivas definições, o que facilita a interpretação de seu conteúdo, por outra pessoa ou outro *software* aplicativo, o que dispensa análise de documentos de referência. Através do XML pode, portanto, estabelecer-se padrões de documentos para troca de informações entre sistemas de processamento, tais como transacções de comércio electrónico, cadeias de provisão, e mais especificamente, informações relativas a processos de negócio ou fluxos de trabalho, entre outras aplicações [33].

### **Web Service Description Language (WSDL)**

Um documento WSDL define um *Schema* XML para descrever um *Web Service*. Os fornecedores de serviços (*Service Providers*) podem descrever os seus próprios *Web Services* através do documento WSDL. Dessa forma, os clientes podem facilmente obter informações sobre os serviços que usarão, tais como: operações, os parâmetros de entrada e saída e ainda a forma como um cliente deve aceder ao serviço. Além de descrever o conteúdo das mensagens, o WSDL define também o endereço onde o serviço está disponível e o protocolo que terá de ser utilizado para comunicar com o serviço. Desta forma, o WSDL possui literalmente toda a informação necessária para que um sistema possa interagir com o *Web Service* [33].

### **JBoss**

O JBoss [34] é um servidor de aplicação de código fonte aberto baseado na plataforma J2EE implementada completamente na linguagem de programação Java. Como é baseada em Java, o JBoss pode ser usado em qualquer Sistema Operacional que suporte Java. O desenvolvimento do JBoss começou em março de 1999. Nascido como um simples container EJB e, ao longo dos anos, evoluiu para ser um servidor de aplicações Java completo, que hoje está bastante maduro. Servidores de aplicação permitem o desenvolvimento de aplicações distribuídas multi-camadas e agem como a interface entre os clientes e as bases de dados e os sistemas de informação corporativos.

### **J2EE**

Java EE (ou J2EE, ou Java 2 *Enterprise Edition*) [35] é uma plataforma de programação de computadores que faz parte da plataforma Java. Voltada para aplicações multi-camadas, baseadas em componentes que são executados em servidor de aplicação, como p.e. JBOSS. A plataforma Java EE é considerada um padrão de desenvolvimento já que o fornecedor de software nesta plataforma deve seguir determinadas regras se quiser declarar os seus produtos como compatíveis com Java EE. Contém bibliotecas desenvolvidas para o acesso a base de dados, RPC, CORBA, etc. Devido a essas

características a plataforma é utilizada principalmente para o desenvolvimento de aplicações corporativas.

A utilização de tecnologia Java e do JBOSS torna o *software* aplicacional independente do sistema operativo. Normalmente os fornecedores de *Application Servers* J2EE, disponibilizam versões para os sistemas operativos correntes (MS Windows, Linux, HP-UX e Solaris). Os exemplos apresentados mais à frente utilizam o sistema operativo Linux RedHat 9.2.

### **MySQL**

O MySQL é um sistema de gestão de base de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language*) como interface. Serve neste caso para manter requisitos mínimos de persistência de algumas entidades, nomeadamente da entidade *Order*.

## **5.8. Pontos relevantes da arquitectura apresentada**

Resumindo, os pontos mais relevantes da arquitectura do Network Activator são:

- Suporte de activação de multi-serviço/multi-recurso. Cada serviço pode ser composto por vários recursos lógicos.
- Interface *northbound web services*, independente do serviço RFS. Esta independência permite que possam ser introduzidos novos serviços sem necessidade de se alterar a interface.
- A interface *northbound* disponibiliza operações síncronas e assíncronas. As primeiras são adequadas para processos de activação com requisitos de tempo real. As segundas permitem o envio na mesma ordem, de uma sequência de serviços, permitindo implementar a funcionalidade de *mass/bulk provisioning*. É ainda possível programar a data/hora de execução de uma ordem.
- Arquitectura modular que separa as funcionalidades de tratamento de pedidos, serviços e recursos, em módulos diferentes, que comunicam entre si por interfaces

bem definidas. Este desenho torna o sistema mais robusto, de modo a que alterações num módulo particular não tenham impacto nos outros módulos do sistema.

- A arquitectura permite a implementação de um sistema flexível, e altamente configurável, em que o suporte de um novo serviço ou recurso (*plug-in*) pode ser realizado sem necessidade de desenvolvimento de software aplicacional.
- O núcleo do sistema é independente do recurso a activar, apenas o *plug-in* detém o conhecimento do mapeamento da informação veiculada na entidade recurso lógico (LR), para os comandos a executar no recurso (NE/EMS) propriamente dito.
- Disponibilidade de uma API para desenvolvimento de *plug-ins* específicos, os quais podem ser instalados em runtime.

## 5.9. Exemplo de configuração de Serviço numa rede MPLS

O exemplo que será apresentado de seguida faz parte do acordo entre a PTInovação e a PTComunicações, para a criação de acessos VPLS *Triple Play Unicast* na sua rede IP/MPLS. Será apresentado o processo geral de configuração do *Network Activator* para suportar novos serviços (configuração de RFS) e novos recursos (configuração de *plug-in*), assim como o fluxo completo de execução da criação de acessos.

### 5.9.1. Configuração do Serviço

Os serviços suportados pelo NA são registados num ficheiro XML (*ServiceDescriptor.xsd*) situado no repositório de ficheiros do sistema, onde são registados os RFSs, como se mostra a seguir de uma forma textual.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include schemaLocation="/xsd/SAM-O.xsd"/>
  <xs:element name="service-descriptor">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
```

```

<xs:element ref="rfsVplsAccess" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Neste caso, o RFS tem o nome de “rfsVplsAccess”, como se pode ver na Figura 29, mostrando também o recurso lógico LR, que compõe o serviço.

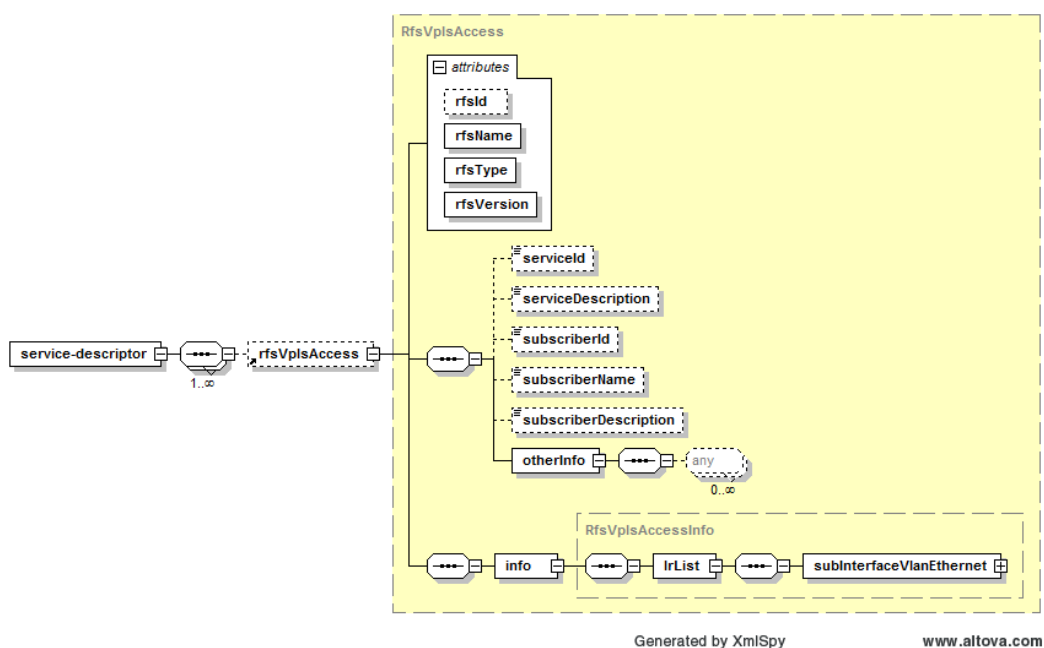


Figura 29 – Composição do RFS rfsVplsAccess

Neste caso, o LR que compõe o serviço é do tipo subInterfaceVlanEthernet, que representa uma subinterface lógica criada sobre uma interface física Ethernet (Figura 30).

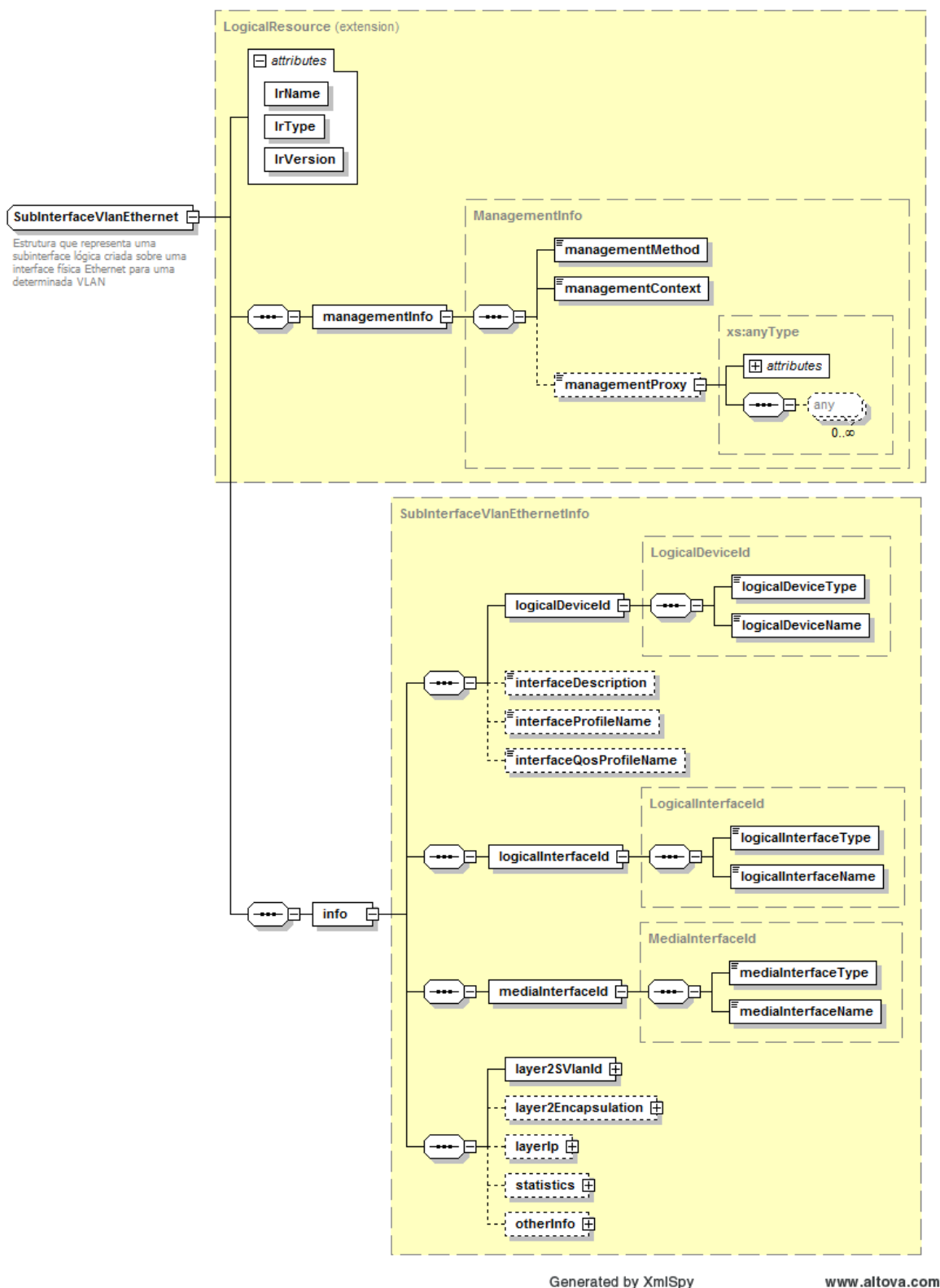


Figura 30 – Representação do recurso lógico subInterfaceVlanEthernet.

No repositório existem dois directórios: um directório chamado xsd, onde são guardados os *schemas* dos documentos XML, e um directório chamado xslt, onde são

guardadas as *scripts* de transformação dos documentos de entrada no sistema. Dentro de cada um desses directórios existe um subdirectório chamado *plugin*, e dentro deste existe outro subdirectório para cada tipo de *plug-in* genérico. Os nomes dos subdirectórios devem ser os nomes dos *plug-ins* genéricos, que é também a *string* passada no atributo *managementMethod* (elemento contido em *managementInfo*) do *Logical Resource*. A figura 30 seguinte ilustra a estrutura do repositório.

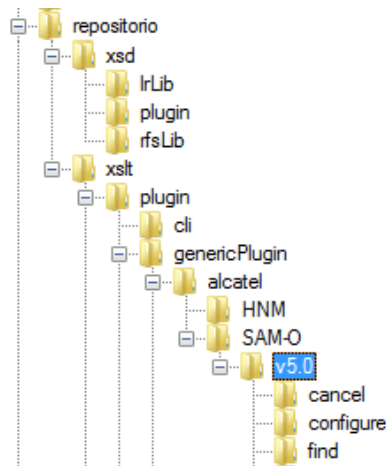


Figura 31 – Estrutura de repositórios do sistema.

A segunda parte do caminho para o ficheiro XSLT é dada pelo campo *managementContext* (elemento contido em *managementInfo*) do *Logical Resource*, geralmente fabricante/modelo/versão.

### 5.9.2. Configuração do Recurso

Com a definição dos RFS e LR's correspondentes aos serviços que se queiram criar, o NA está pronto a receber pedidos através da interface WebService disponibilizada aos seus possíveis clientes, entre os quais sistemas de gestão de serviços ou portais de internet, por exemplo.

Foi implementado um cliente por linha de comandos, que usa a API Apache Axis [37] que usa a interface WebService disponibilizada pelo NA. O cliente recebe os seguintes parâmetros:

```
[naom@PCDPIN01 om]$ ./activate.sh
```

```
usage: om.process <sincrono> <xml info> <descricao> <prioridade> <operacao>
```

```
<sincrono>: 1, 0
```

```
<xml info>: Ficheiro xml com a ordem
```

```
<descricao>: Descricao da ordem
```

```
<prioridade>: 1, 2, 3, 4
```

```
<operacao>: Conjunto de caracteres que define o campo 'operation' da ordem
```

No caso prático aqui apresentado, que se baseia na criação de acessos VPLS Triple Play Unicast, o documento a enviar ao NA terá a estrutura apresentada de seguida:

```
<?xml version="1.0" encoding="UTF-8"?>
<service-descriptor xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions">
  <rfsVplsAccess rfsType="type1" rfsName="rfsVplsAccess" rfsVersion="v1.0">
    <serviceId>1009510002</serviceId>
    <subscriberId>desconhecido</subscriberId>
    <subscriberName>desconhecido</subscriberName>
    <subscriberDescription>desconhecido</subscriberDescription>
    <otherInfo/>
    <info>
      <lrList>
        <subInterfaceVlanEthernet lrVersion="v1.0" lrType="type1"
lrName="subInterfaceVlanEthernet">
          <managementInfo>
            <managementMethod>genericPlugin</managementMethod>
            <managementContext>alcatel/SAM-O/v5.0</managementContext>
          </managementInfo>
          <info>
            <logicalDeviceId>
              <logicalDeviceType>ROUTER</logicalDeviceType>
              <logicalDeviceName>SR12</logicalDeviceName>
            </logicalDeviceId>
            <interfaceDescription>150177777-288888888</interfaceDescription>
            <interfaceQosProfileName>QOS0</interfaceQosProfileName>
            <logicalInterfaceId>
              <logicalInterfaceType>SubEthernet</logicalInterfaceType>
              <logicalInterfaceName>11/71</logicalInterfaceName>
            </logicalInterfaceId>
            <mediaInterfaceId>
              <mediaInterfaceType>GigabitEthernet</mediaInterfaceType>
              <mediaInterfaceName>LAG-20</mediaInterfaceName>
            </mediaInterfaceId>
            <layer2SVlanId>
              <oVlan>0</oVlan>
            </layer2SVlanId>
            <otherInfo>
              <servicoComercial>IPTV-UnicastV4</servicoComercial>
              <classeTecnica>T60</classeTecnica>
              <nrSetTopBoxes>2</nrSetTopBoxes>
              <classeComercial>92</classeComercial>
            </otherInfo>
          </info>
        </subInterfaceVlanEthernet>
      </lrList>
    </info>
  </rfsVplsAccess>
</service-descriptor>
```



```

<telefoneCliente>288888888</telefoneCliente>
<nrServicoADSL>150177777</nrServicoADSL>
<dRegisto>02-02-2008 5:56:52</dRegisto>
<nomeCliente>PAULO E HELENA , LDA</nomeCliente>
<DSLAM>63AV01/02</DSLAM>
<portDSLAM>11/71</portDSLAM>
<slot/>
<operador>PTC</operador>
<encaps>IPoE</encaps>
<ipFixo>D</ipFixo>
<recursoDSLAM>1/1/4/40</recursoDSLAM>
<servico>2</servico>
<operacao>15</operacao>
<ptr1Caract>-</ptr1Caract>
<destinoPortoTipo>LAG</destinoPortoTipo>
<DSLAMFabricante>Alcatel</DSLAMFabricante>
<versaoBBRAS>ALCATEL/-</versaoBBRAS>
<stbCaract>desconhecido</stbCaract>
<pacotesTV>Pack trial</pacotesTV>
<serviceId>1009510002</serviceId>
<subscriberId>desconhecido</subscriberId>
<esmPolicy/>
<numLeases/>
</otherInfo>
</info>
</subInterfaceVlanEthernet>
</lrList>
</info>
</rfsVplsAccess>
</service-descriptor>

```

O comando inserido no cliente será, para o caso prático apresentado:

```
[PCDPIN01 om]$ ./activate.sh 1 AccessUnicastConfigure.xml "PedidoConfigure" 1 configure
```

Este é um exemplo de um pedido enviado ao NA para configuração de um acesso Unicast no sistema de gestão da PTComunicações. Mais à frente irá ser apresentado o documento enviado efectivamente para o sistema de gestão.

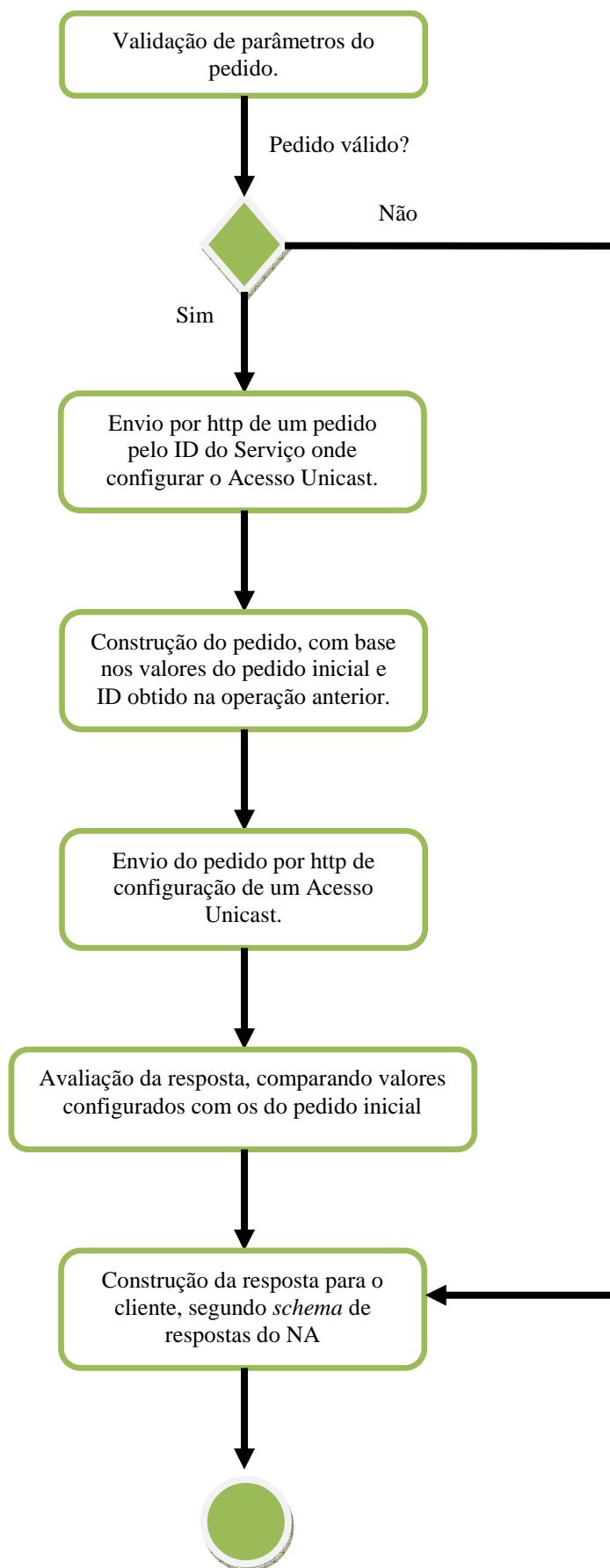
Quando este pedido chega ao NA, é aplicada uma transformação inicial, que agrupa todos os LRs do mesmo pedido. Após esta transformação inicial, é aplicada a seguinte regra para obtenção da *script* que irá transformar o pedido em comandos ou outros documentos XML que serão enviados para os equipamentos ou para outros sistemas de gestão:

*repositorio/xslt/managementMethod/managementContext/rfsName/operation/ lrName\_lrVersion.xslt*

Neste exemplo, teríamos o seguinte path:

*repositorio/xslt/genericPlugin/alcatel/SAM-O/v5.0/rfsVplsAccess/configure/subInterfaceVlanEthernet\_v1.0.xslt*

A constituição desta *script* de transformação é apresentada no Anexo 1. Esta transformação acontece sempre que um pedido é enviado para o NA e é dependente dos valores enviados, ou seja, a uma operação de configuração, está associada uma determinada script, a uma operação de cessação está associada outra script e assim sucessivamente. Neste caso prático, a script de transformação irá construir o seguinte fluxo de configurações:



Após configuração do Acesso Unicast, o NA envia um relatório de resposta, informando o sucesso ou insucesso da operação. Esse relatório é apresentado no Anexo II. No Anexo III, está a configuração do acesso criado, obtido com uma operação de pesquisa ao serviço MPLS em questão.

# **Capítulo 6: Conclusões e trabalho futuro.**

Na actual conjuntura do mercado das telecomunicações, parece claro que o futuro dos grandes operadores e prestadores de serviços passa pela convergência das redes actualmente existentes numa única infra-estrutura em que a tecnologia e serviços IP desempenham um papel fundamental. Nestas circunstâncias, onde as empresas são fortemente pressionados para a necessidade de uma actualização devido às novas tendências, para a forte competitividade e para o alto grau de exigência ao serviço do cliente final, é imprescindível contar com uma excelente infra-estrutura de sistemas de gestão, perfeitamente sincronizada, em que a execução automática dos processos de negócio ocupe virtualmente os cem por cento das tarefas de operação, gestão, exploração e manutenção das novas redes IP, de modo que se consigam níveis máximos de eficiência e redução de custos de manutenção, ao mesmo tempo que se aumentam os níveis de satisfação dos clientes finais.

Além disso, a necessidade de dar respostas rápidas ao aparecimento de novos serviços e tecnologias, como é o caso do serviço de voz, vídeo e televisão sobre IP, juntamente com o aumento constante de clientes que aspiram aceder às novas tecnologias de informação, fazem com que estas arquitecturas de sistemas de gestão devam apresentar determinadas características quanto à capacidade de crescimento e flexibilidade para a introdução de novos processos. Neste contexto, a arquitectura NGOSS defendida pelo TMForum, parece adaptar-se a todos estes princípios e necessidades, e conta com os mais altos níveis de aceitação relativamente a outras iniciativas também existentes no mercado.

Existe actualmente um grande número de fornecedores de sistemas de gestão BSS e OSS que se adaptam em maior ou menor extensão aos princípios defendidos pelo NGOSS. Este número de fornecedores, vai sendo reduzido gradualmente através de um processo de selecção natural, onde obviamente, irá perdurar aquele que alcançar uma posição mais

sólida e saiba adaptar-se melhor às novas necessidades de operadores e prestadores de serviços.

Neste momento, as companhias de telecomunicações vêm-se imersas num dilema para decidir o caminho a tomar na altura de implementar uma arquitectura OSS, podendo decidir entre evoluir para o uso de plataformas comerciais sempre que isto seja possível, tal como defende o NGOSS, que é a tendência predominante neste momento, ou continuar a desenvolver soluções à medida. O investimento necessário para a implementação destas arquitecturas tem um papel importante nesta decisão, mas o custo da aquisição destes novos produtos não é o único a considerar, já que é necessário contemplar outros factores que preocupam as empresas, tais como os itens necessários à integração das plataformas, entre si e com as plataformas já existentes, o custo associado à migração dos dados e processos para os novos sistemas, e o custo da manutenção e da operação das arquitecturas resultantes [36].

Além disso, é necessário ter sempre em consideração alguma dúvidas de que os novos sistemas permitem alcançar melhores resultados que os actuais. Finalmente não há que desprezar os investimentos necessários para tornar a exploração inicial dos componentes candidatos. Por todas estas razões, a adopção, em primeiro lugar, por parte dos fabricantes de software da plataforma J2EE como uma ferramenta de desenvolvimento que assegura níveis óptimos em termos de capacidade de crescimento, flexibilidade e eficiência, e a subsequente incorporação dos fundamentos da iniciativa OSS/J como veículo de integração entre sistemas, pode ser um factor diferencial que actue sobre as operadoras e os prestadores de serviços para que haja uma aprovação definitiva das plataformas BSS e OSS destas características.

Não obstante, e embora pareça claro que finalmente o uso dos sistemas comerciais (COTS) prevalecerá, existirão sempre factores e necessidades específicas nessas empresas, embora cada vez em menor número, que força a que seja impossível não contar com determinados elementos que não podem ser consideradas COTS e são o resultado de desenvolvimentos feitos sob medida.

Além disso deve-se ter em conta que a adopção uma estratégia OSS/J não dificulta, mas facilita a incorporação de alguns componentes do desenvolvimento sob medida para satisfazer as necessidades específicas das operadoras. Estes componentes só terão de se ajustar ao padrão que representa o OSS/J.

No âmbito dos processos de provisão de serviços (*Fullfilment*), o processo *Service Activation*, tem a responsabilidade de executar os comandos de configuração nos recursos (NE/EMS), de forma a concretizar a activação do serviço. O *Network Activator* pretende ser um sistema de informação que suporta este processo, posicionando-se na arquitectura de referência (Figura 18) como sendo uma Plataforma de Activação de Serviços Multi-Serviço. Neste sentido, o NA possibilita a provisão automática e a reconfiguração de recursos em tempo real, de a forma a satisfazer os pedidos de serviço provenientes de diferentes plataformas de *Order Management e Serviços* (SAPA, Plataforma Wi-Fi, Plataforma IN, etc), via *bus* integrado de comunicações. Esta flexibilidade permite a provisão automática de serviços end-to-end, e a alteração por parte dos clientes de regras e serviços definidos para o seu acesso, dinamizando a criação de novas ofertas e possibilitando a aplicação de novos modelos de negócio, dando mais valor aos clientes.

Tendo em consideração o enquadramento do *Network Activator* e os requisitos identificados, a arquitectura do sistema foi desenhada com o objectivo de permitir a implementação de um sistema flexível, e configurável, em que sempre que possível, o suporte de um novo serviço ou recurso (*plug-in*), se resuma à construção de *scripts XML*, sem necessidade de desenvolvimento de *software* aplicacional.

A arquitectura proposta apresenta os seguintes pontos relevantes:

- Segue os princípios da arquitectura NGOSS, de interligação de sistemas de OSS/BSS, definidos no *TMForum*. Neste sentido o *Network Activator*, está preparado para poder interligar-se num bus comum de comunicação entre sistemas, usando a tecnologia *web services*.
- Utilização do modelo SID, do *TMForum*,
- A interface northbound disponibilizada é agnóstica ao serviço, permitindo que sejam introduzidos novos serviços sem necessidade de se alterar a interface. Esta interface é uma adaptação da *API Service Activation*, da iniciativa OSS/J, para *web services*.
- A interface *northbound* disponibiliza operações síncronas e assíncronas. As primeiras são adequadas para processos de activação com requisitos de tempo real. As segundas permitem o envio na mesma ordem, de uma sequência de serviços, permitindo implementar a funcionalidade de *mass/bulk provisioning*.
- Arquitectura modular que separa as funcionalidades de tratamento de pedidos, serviços e recursos, em módulos diferentes, que comunicam entre si por interfaces

bem definidas. Este desenho torna o sistema mais robusto, de forma a que alterações num módulo particular não tenham impacto nos outros módulos do sistema.

- O núcleo do sistema é independente do recurso a activar, apenas o *plug-in* detém o conhecimento do mapeamento da informação veiculada na entidade recurso lógico (LR), para os comandos a executar no recurso (NE/EMS) propriamente dito.
- Suporte de plug-ins genéricos, tipicamente orientados ao protocolo (CLI, SNMP, http, FTP, etc), configuráveis por *scripts*. A configuração de um recurso resume-se à construção de *scripts* XSL e ao seu registo no *Plug-in Registrar*.
- Disponibilidade de uma API para desenvolvimento de *plug-ins* específicos, os quais podem ser instalados em *runtime*.
- Arquitectura física com garantia de desempenho, escalabilidade e redundância, é baseada nas mais recentes tecnologias de sistemas de informação, nomeadamente:
  - Middleware J2EE (EJB + JMS), no *core* das plataformas,
  - XML, na definição de services, recursos, interfaces e ficheiros/*scripts* de configuração.

Em termos de perspectivas futuras, prevê-se a aplicação concreta do *Network Activator* aos serviços MEO da PTCom, assim como a gestão de equipamentos na rede ViVo no Brasil.



# Anexo I

Constituição do ficheiro XSLT de transformação *subInterfaceVlanEthernet\_v1.0.xslt*

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:import href="/xslt/plugin/genericPlugin/commons.xslt"/>
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>

  <xsl:variable name="catalogo"
select="document('/xslt/plugin/CatalogoEquipsSAM.xml')/equipamentos"/>
  <xsl:variable name="xsltTransformationFilePath_configure" select="
'/home/na/repositorio/xslt/plugin/genericPlugin/alcatel/SAM-O/v5.0/configure' "/>
  <xsl:variable name="xsltTransformationFileName_configure" select="
'xsltTransform_configure.xslt' "/>

  <!-- ELEMENTOS OBRIGATORIOS -->
  <xsl:variable name="equipmentName"
select="/lr/info/logicalDeviceId/logicalDeviceName/text()"/>
  <xsl:variable name="nrServicoADSL" select="/lr/info/otherInfo/nrServicoADSL/text()"/>
  <xsl:variable name="telefoneCliente" select="/lr/info/otherInfo/telefoneCliente/text()"/>
  <xsl:variable name="idVPLS" select="/lr/rfs/serviceId/text()"/>
  <xsl:variable name="servicoComercial" select="/lr/info/otherInfo/servicoComercial/text()"/>
  <xsl:variable name="nrSetTopBoxes" select="/lr/info/otherInfo/nrSetTopBoxes/text()"/>
  <xsl:variable name="logicalInterfaceName"
select="/lr/info/logicalInterfaceId/logicalInterfaceName/text()"/>
  <xsl:variable name="mediaInterfaceName"
select="/lr/info/mediaInterfaceId/mediaInterfaceName/text()"/>

  <!-- ELEMENTOS NAO OBRIGATORIOS -->
  <xsl:variable name="interfaceDescription" select="/lr/info/interfaceDescription/text()"/>
  <xsl:variable name="subscriberId" select="/lr/rfs/subscriberId/text()"/>
  <xsl:variable name="subscriberName" select="/lr/rfs/subscriberName/text()"/>
  <xsl:variable name="subscriberDescription" select="/lr/rfs/subscriberDescription/text()"/>
  <xsl:variable name="classeTecnica" select="/lr/info/otherInfo/classeTecnica/text()"/>
  <xsl:variable name="classeComercial" select="/lr/info/otherInfo/classeComercial/text()"/>

  <!-- OBTIDOS -->
  <xsl:variable name="logicalDeviceName"
select="$catalogo/linha[EquipmentName=nomeEquipamento]/ipEquipamento"/>
  <xsl:variable name="isBloqueado"
select="$catalogo/linha[nomeEquipamento=$equipmentName]/adminStatus"/>
  <xsl:variable name="innerVlan" select="substring-after($logicalInterfaceName, '/')"/>
  <xsl:variable name="outerVlan" select="substring-before($logicalInterfaceName, '/')"/>

  <xsl:variable name="idQoS-Ingress">37000</xsl:variable>
  <xsl:variable name="idQoS-Egress"><xsl:value-of
select="concat('370', $classeComercial)"/></xsl:variable>

  <xsl:template name="isToSkip">
    <xsl:param name="logicalDeviceName"/>
    <xsl:param name="isBloqueado"/>
    <xsl:param name="nrServicoADSL"/>
    <xsl:param name="telefoneCliente"/>
    <xsl:param name="idVPLS"/>
```

---

```

<xsl:param name="servicoComercial"/>
<xsl:param name="logicalInterfaceName"/>

<xsl:choose>
  <!-- Verifica se o logicalDeviceName(SR) está presente no catalogo local-->
  <xsl:when test="not(string($logicalDeviceName))">
    <skipConfigMessage code="MPLS_002">The Switch Router '<xsl:copy-of
select="$equipmentName"/>' does not exist in the SR Catalog</skipConfigMessage>
  </xsl:when>
  <!-- Verifica se o logicalDeviceName(SR) está bloqueado-->
  <xsl:when test="$isBloqueado='bloqueado'">
    <skipConfigMessage code="MPLS_003">The Switch Router '<xsl:copy-of
select="$equipmentName"/>' is blocked in the SR Catalog</skipConfigMessage>
  </xsl:when>
  <!-- Verifica a existencia do elemento nrServicoADSL-->
  <xsl:when test="not(string($nrServicoADSL))">
    <skipConfigMessage code="MPLS_004">The element AdslServiceNumber must not
be empty</skipConfigMessage>
  </xsl:when>
  <!-- Verifica o formato do elemento nrServicoADSL-->
  <xsl:when test="starts-with($nrServicoADSL,'500')">
    <skipConfigMessage code="MPLS_005">The ADSL number '<xsl:copy-of
select="$nrServicoADSL"/>' must not begin with 500xxxxxx</skipConfigMessage>
  </xsl:when>
  <!-- Verifica a existencia do elemento telefoneCliente-->
  <xsl:when test="not(string($telefoneCliente))">
    <skipConfigMessage code="MPLS_006">The element phoneClient must not be
empty</skipConfigMessage>
  </xsl:when>
  <!-- Verifica o formato do elemento idVPLS-->
  <xsl:when test="not(starts-with($idVPLS,'100951000'))">
    <skipConfigMessage code="MPLS_007">The VPLS id must begin with
100951000</skipConfigMessage>
  </xsl:when>
  <!-- Verifica o formato do elemento servicoComercial-->
  <xsl:when test="not(contains($servicoComercial,'IPTV-Unicast') or
contains($servicoComercial,'IPTV-Multicast'))">
    <skipConfigMessage code="MPLS_008">The comercial service '<xsl:copy-of
select="$servicoComercial"/>' is invalid. (use IPTV-Unicast or IPTV-
Multicast)</skipConfigMessage>
  </xsl:when>
  <!-- Verifica o formato do elemento logicalInterfaceName -->
  <xsl:when test="not(contains($logicalInterfaceName,'/'))">
    <skipConfigMessage code="MPLS_009">The element logicalInterfaceName must
contain "innerVlan/outerVlan"</skipConfigMessage>
  </xsl:when>
</xsl:choose>
</xsl:template>

<xsl:template match="/">
  <pluginResource>
    <xsl:copy-of select="$lrDetailsElement"/>
    <xsl:copy-of select="$rfsDetailsElement"/>
    <variableList>
      <variable name="receivedLr">
        <xml><xsl:copy-of select="/lr[@lrName=$lrName]"/></xml>
      </variable>
      <variable name="descriptionConfigurada"/>
      <variable name="servicIdValidado"/>
    </variableList>
  </pluginResource>
</xsl:template>

```

---

```

<variable name="classeTecnica">
  <string><xsl:copy-of select="$classeTecnica"/></string>
</variable>
<variable name="classeComercial">
  <string><xsl:copy-of select="$classeComercial"/></string>
</variable>
<variable name="equipmentName">
  <string><xsl:copy-of select="$equipmentName"/></string>
</variable>
<variable name="httpPost_findSAP_Resposta"/>
<variable name="httpTask_askAccessInfo">
  <boolean>>false</boolean>
</variable>
<variable name="terminouXsltTransform_avaliaoAcessoConfigurado">
  <boolean>>false</boolean>
</variable>
<variable name="primeiraEntrada">
  <string>primeiraEntrada</string>
</variable>
<variable name="segundaEntrada">
  <string>segundaEntrada</string>
</variable>
<variable name="httpTask_sendAccessConfigure">
  <boolean>>false</boolean>
</variable>
<variable name="nrSetTopBoxes">
  <string><xsl:copy-of select="$nrSetTopBoxes"/></string>
</variable>
<variable name="nrServicoADSL">
  <string><xsl:copy-of select="$nrServicoADSL"/></string>
</variable>
<variable name="telefoneCliente">
  <string><xsl:copy-of select="$telefoneCliente"/></string>
</variable>
<variable name="innerVlan">
  <string><xsl:copy-of select="$innerVlan"/></string>
</variable>
<variable name="outerVlan">
  <string><xsl:copy-of select="$outerVlan"/></string>
</variable>
<variable name="mediaInterfaceName">
  <string><xsl:copy-of select="$mediaInterfaceName"/></string>
</variable>
<variable name="idQoS-Ingress">
  <string><xsl:copy-of select="$idQoS-Ingress"/></string>
</variable>
<variable name="idVPLS">
  <string><xsl:copy-of select="$idVPLS"/></string>
</variable>
<variable name="servicoComercial">
  <string><xsl:copy-of select="$servicoComercial"/></string>
</variable>
<variable name="logicalDeviceName">
  <string><xsl:value-of select="$logicalDeviceName"/></string>
</variable>
<variable name="httpTask_findCorrectServiceId">
  <boolean>>false</boolean>
</variable>
<variable name="terminouXsltTransform_scriptConfigureAccess">
  <boolean>>false</boolean>

```

---

```

</variable>
<variable name="doTasks">
  <boolean>true</boolean>
</variable>
<variable name="samLogin">
  <string><xsl:copy-of select="$samLogin"/></string>
</variable>
<variable name="samPassword">
  <string><xsl:copy-of select="$samPassword"/></string>
</variable>
<variable name="httpPost_AccessConfigure"/>
<variable name="httpPost_AccessConfigure_Resposta"/>
<variable name="httpPost_findServiceId_Resposta"/>
<variable name="httpPost_findServiceId">
  <string><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <header xmlns="xmlapi_1.0">
      <security>
        <user>]]><xsl:copy-of select="$samLogin"/><![CDATA[</user>
        <password>]]><xsl:copy-of select="$samPassword"/><![CDATA[</password>
      </security>
      <requestID>NA-FindServiceId ]]><xsl:copy-of select="$idVPLS"/><![CDATA[</requestID>
    </header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <find xmlns="xmlapi_1.0">
      <fullClassName>service.Service</fullClassName>
      <filter><equal name="serviceId" value="]]><xsl:value-of
select="$idVPLS"/><![CDATA["/></filter>
      <resultFilter>
        <attribute>objectFullName</attribute>
        <attribute>id</attribute>
        <children/>
      </resultFilter>
    </find>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>]]></string>
</variable>
<variable name="httpPost_findSAP"/>
</variableList>
<returnValues>
  <variable>idVPLS</variable>
  <variable>nrSetTopBoxes</variable>
  <variable>classeTecnica</variable>
  <variable>classeComercial</variable>
  <variable>servicoComercial</variable>
  <variable>mediaInterfaceName</variable>
  <variable>outerVlan</variable>
  <variable>innerVlan</variable>
  <variable>telefoneCliente</variable>
  <variable>nrServicoADSL</variable>
  <variable>equipmentName</variable>
  <variable>httpPost_AccessConfigure_Resposta</variable>
  <variable>httpPost_findSAP_Resposta</variable>
  <variable>descriptionConfigurada</variable>
  <variable>receivedLr</variable>
</returnValues>
<flowControl/>
<sessionManagement>

```

---

```

<xsl:attribute name="id"><xsl:value-of select="$lId"/></xsl:attribute>
<init>
  <flowControl/>
  <tasks/>
</init>
<terminate>
  <flowControl/>
  <tasks/>
</terminate>
<variableList/>
<returnValues/>
</sessionManagement>
<config>
  <!-- ===== Procura id do ServiceID do pedido pois VPLS pode ter ser
criada pelo SAM e nao pelo NA, logo o idVPLS pode ser diferente =====-->
  <task name="http">
    <variableList/>
    <flowControl>
      <executed>httpTask_findCorrectServiceId</executed>
    </flowControl>
    <forceNewSession>0</forceNewSession>
    <forceCloseSession>0</forceCloseSession>
    <session>
      <host><xsl:value-of select="$samHost"/></host>
      <port><xsl:value-of select="$samPort"/></port>
      <file><xsl:value-of select="$samFile"/></file>
    </session>
    <action>
      <post varName="httpPost_findServiceId_Resposta" connectionTimeout="1"
readTimeout="1" replyType="xml">
        <varName processingInstruction="false">httpPost_findServiceId</varName>
      </post>
    </action>
  </task>
  <!-- ===== Construção do xml de pedido VPLS Access ===== -->
  <task name="xsltTransform">
    <variableList/>
    <flowControl>
      <semaphore>httpTask_findCorrectServiceId</semaphore>
      <executed>terminouXsltTransform_scriptConfigureAccess</executed>
    </flowControl>
    <forceNewSession>0</forceNewSession>
    <forceCloseSession>0</forceCloseSession>
    <xsltPath>
      <sourceFileName><xsl:copy-of
select="$xsltTransformationFileName_configure"/></sourceFileName>
      <sourceDir><xsl:copy-of
select="$xsltTransformationFilePath_configure"/></sourceDir>
    </xsltPath>
    <includeVarList>
      <varName>primeiraEntrada</varName>
      <varName>httpPost_findServiceId_Resposta</varName>
      <varName>httpPost_AccessConfigure</varName>
      <varName>doTasks</varName>
      <varName>idVPLS</varName>
      <varName>servicoComercial</varName>
      <varName>logicalDeviceName</varName>
      <varName>idQoS-Ingress</varName>
      <varName>mediaInterfaceName</varName>
      <varName>samLogin</varName>
    </includeVarList>
  </task>
</config>

```

---

```

        <varName>samPassword</varName>
        <varName>innerVlan</varName>
        <varName>outerVlan</varName>
        <varName>telefoneCliente</varName>
        <varName>nrServicoADSL</varName>
        <varName>descriptionConfigurada</varName>
    </includeVarList>
</task>
<!-- ===== Envio do pedido de configuracao httpPost_AccessConfigure =====>
    <task name="http">
        <variableList/>
        <flowControl>
            <semaphore>terminouXsltTransform_scriptConfigureAccess</semaphore>
            <executed>httpTask_sendAccessConfigure</executed>
        </flowControl>
        <forceNewSession>0</forceNewSession>
        <forceCloseSession>0</forceCloseSession>
        <session>
            <host><xsl:value-of select="$samHost"/></host>
            <port><xsl:value-of select="$samPort"/></port>
            <file><xsl:value-of select="$samFile"/></file>
        </session>
        <action>
            <post varName="httpPost_AccessConfigure_Resposta" connectionTimeout="1"
readTimeout="1" replyType="xml">
                <varName
processingInstruction="false">httpPost_AccessConfigure</varName>
            </post>
        </action>
    </task>
    <!-- ===== Avalia se já existe objecto criado, de modo a obter os parametros
configurados e comparar com os do pedido ===== -->
    <task name="xsltTransform">
        <variableList/>
        <flowControl>
            <semaphore>httpTask_sendAccessConfigure</semaphore>
            <executed>terminouXsltTransform_avaliaAcessoConfigurado</executed>
        </flowControl>
        <forceNewSession>0</forceNewSession>
        <forceCloseSession>0</forceCloseSession>
        <xsltPath>
            <sourceFileName><xsl:copy-of
select="$xsltTransformationFileName_configure"/></sourceFileName>
            <sourceDir><xsl:copy-of
select="$xsltTransformationFilePath_configure"/></sourceDir>
        </xsltPath>
        <includeVarList>
            <varName>segundaEntrada</varName>
            <varName>httpPost_AccessConfigure_Resposta</varName>
            <varName>doTasks</varName>
            <varName>idVPLS</varName>
            <varName>httpPost_findSAP</varName>
            <varName>samLogin</varName>
            <varName>samPassword</varName>
            <varName>innerVlan</varName>
            <varName>outerVlan</varName>
            <varName>mediaInterfaceName</varName>
            <varName>logicalDeviceName</varName>
        </includeVarList>
    </task>

```

---

```

<!-- ===== Envio do pedido de info do acesso ja configurado
=====-->
<task name="http">
  <variableList/>
  <flowControl>
    <semaphore>terminouXsltTransform_avaliaAcessoConfigurado</semaphore>
    <toDo>doTasks</toDo>
    <executed>httpTask_askAccessInfo</executed>
  </flowControl>
  <forceNewSession>0</forceNewSession>
  <forceCloseSession>0</forceCloseSession>
  <session>
    <host><xsl:value-of select="$samHost"/></host>
    <port><xsl:value-of select="$samPort"/></port>
    <file><xsl:value-of select="$samFile"/></file>
  </session>
  <action>
    <post varName="httpPost_findSAP_Resposta" connectionTimeout="1"
readTimeout="1" replyType="xml">
      <varName processingInstruction="false">httpPost_findSAP</varName>
    </post>
  </action>
</task>
</config>
<xsl:call-template name="isToSkip">
  <xsl:with-param name="logicalDeviceName" select="$logicalDeviceName"/>
  <xsl:with-param name="isBloqueado" select="$isBloqueado"/>
  <xsl:with-param name="nrServicoADSL" select="$nrServicoADSL"/>
  <xsl:with-param name="telefoneCliente" select="$telefoneCliente"/>
  <xsl:with-param name="idVPLS" select="$idVPLS"/>
  <xsl:with-param name="servicoComercial" select="$servicoComercial"/>
  <xsl:with-param name="logicalInterfaceName" select="$logicalInterfaceName"/>
</xsl:call-template>
</pluginResource>
</xsl:template>
</xsl:stylesheet>

```



# Anexo II

Resposta em XML do pedido de Acesso Unicast enviado ao Network Activator.

```
<?xml version="1.0" encoding="UTF-8"?>
<rfsResponse rfsName="rfsVplsAccess" rfsType="type1" rfsVersion="v1.0">
  <lrResponseList>
    <lrResponse lrName="subInterfaceVlanEthernet" lrType="type1" lrVersion="v1.0"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xtl="xmlapi_1.0">
      <response code="MPLS_000">Operation Successful</response>
      <description>The Unicast Access with with serviceADSL: '1234567890' e
clientPhoneNumber: '1223456789' was configured correctly</description>
      <data>
        <subInterfaceVlanEthernet lrId="LR_1_RFS92151"
lrName="subInterfaceVlanEthernet" lrType="type1" lrVersion="v1.0">
          <managementInfo xmlns:fn="http://www.w3.org/2005/xpath-
functions" xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <managementMethod>genericPlugin</managementMethod>
            <managementContext>alcatel/SAM-
O/v5.0</managementContext>
            </managementInfo>
            <info>
              <logicalDeviceId>
                <logicalDeviceType>ROUTER</logicalDeviceType>
                <logicalDeviceName>SR12</logicalDeviceName>
                </logicalDeviceId>
                <interfaceDescription>1234567890-
1223456789</interfaceDescription>
                <interfaceQosProfileName>QOS0</interfaceQosProfileName>
                <logicalInterfaceId>
                  <logicalInterfaceType>SubEthernet</logicalInterfaceType>
                  <logicalInterfaceName>11/125</logicalInterfaceName>
                  </logicalInterfaceId>
                  <mediaInterfaceId>
                    <mediaInterfaceType>Ethernet</mediaInterfaceType>
                    <mediaInterfaceName>lag-
11</mediaInterfaceName>
                    </mediaInterfaceId>
                    <layer2SVlanId>
                      <oVlan>0</oVlan>
                      </layer2SVlanId>
                      <otherInfo xmlns:fn="http://www.w3.org/2005/xpath-
functions" xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <servicoComercial>IPTV-
UnicastV4</servicoComercial>
                        <telefoneCliente>1223456789</telefoneCliente>
```



---

```
<nrServicoADSL>1234567890</nrServicoADSL>
                                </otherInfo>
                                </info>
                                </subInterfaceVlanEthernet>
                                </data>
                                </lrResponse>
                                </lrResponseList>
                                </rfsResponse>
```

---

# Anexo III

Constituição do ficheiro XML de resposta ao Acesso Unicast configurado no sistema remoto.

```
<?xml version="1.0" encoding="UTF-8"?>
<rfsResponse rfsName="rfsVplsAccess" rfsType="type1" rfsVersion="v1.0">
  <lrResponseList>
    <lrResponse lrName="subInterfaceVlanEthernet" lrType="type1" lrVersion="v1.0"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xtl="xmlapi_1.0">
      <response code="MPLS_000">Operation Successful</response>
      <description>The Unicast Access was found</description>
      <data>
        <vpls.L2AccessInterface xmlns="xmlapi_1.0">
          <shgSitePointer>svc-mgr:service-1431:10.100.100.2:shg-SHG</shgSitePointer>
          <shgName>SHG</shgName>
          <lightweightSap>true</lightweightSap>
          <ossMetaData>N/A</ossMetaData>
          <egressMcastGroupPointer/>
          <egressMcastGroupName>N/A</egressMcastGroupName>
          <vlanTranslation>None</vlanTranslation>
          <vlanTranslationId>-1</vlanTranslationId>
          <vplsMode>none</vplsMode>
          <redundantServiceId>0</redundantServiceId>
          <encapTagging>untagged</encapTagging>
          <defaultVlan>false</defaultVlan>
          <shcvSourceIpAddress>0.0.0.0</shcvSourceIpAddress>
          <shcvSourceMacAddress>00-00-00-00-00-00</shcvSourceMacAddress>
          <shcvAction>alarm</shcvAction>
          <shcvInterval>0</shcvInterval>
          <cesBackplanePortBindingPointer/>
          <l2ptTermination>disabled</l2ptTermination>
          <l2ptProtocol>
            <bit>stp</bit>
          </l2ptProtocol>
          <bpduTransOper>2</bpduTransOper>
          <bpduTranslation>disabled</bpduTranslation>
          <usesMultipointShared>false</usesMultipointShared>
          <macMonitoring>false</macMonitoring>
          <todSuitePointer/>
          <svcComponentId>1431</svcComponentId>
          <subscriberId>5000</subscriberId>
          <subscriberName>IPTV</subscriberName>
          <serviceId>1009510002</serviceId>
          <serviceName>VPLS 1009510002</serviceName>
          <serviceType>vpls</serviceType>
          <dIconX>0</dIconX>
          <dIconY>0</dIconY>
          <ingressPolicyId>1</ingressPolicyId>
          <ingressPolicyName>Access Ingress-1</ingressPolicyName>
          <ingressPolicyObjectPointer>Access Ingress:1</ingressPolicyObjectPointer>
          <egressPolicyId>37000</egressPolicyId>
        </vpls.L2AccessInterface>
      </data>
    </lrResponse>
  </lrResponseList>
</rfsResponse>
```

---

```

<egressPolicyName>IPTV-UCAST-MARK-DOT1P</egressPolicyName>
<egressPolicyObjectPointer>Access Egress:37000</egressPolicyObjectPointer>
<aggregation>off</aggregation>
<ingressSchedulerObjectPointer/>
<ingressSchedulerName>N/A</ingressSchedulerName>
<egressSchedulerName>N/A</egressSchedulerName>
<egressSchedulerObjectPointer/>
<aggregationSchedulerName>N/A</aggregationSchedulerName>
<aggregationSchedulerObjectPointer/>
<ingressFilterPointer/>
<ingressFilterType>unspecified</ingressFilterType>
<ingressFilterId>0</ingressFilterId>
<ingressFilterName>N/A</ingressFilterName>
<ingressMacFilterId>0</ingressMacFilterId>
<ingressIpFilterId>0</ingressIpFilterId>
<ingressMatchQinqDot1pBits>none</ingressMatchQinqDot1pBits>
<egressQinqMarkTopBitsOnly>>false</egressQinqMarkTopBitsOnly>
<egressFilterPointer/>
<egressFilterType>unspecified</egressFilterType>
<egressFilterId>0</egressFilterId>
<egressFilterName>N/A</egressFilterName>
<egressMacFilterId>0</egressMacFilterId>
<egressIpFilterId>0</egressIpFilterId>
<ingressIpv6FilterPointer/>
<ingressIpv6FilterId>0</ingressIpv6FilterId>
<ingressIpv6FilterName>N/A</ingressIpv6FilterName>
<egressIpv6FilterPointer/>
<egressIpv6FilterId>0</egressIpv6FilterId>
<egressIpv6FilterName>N/A</egressIpv6FilterName>
<accountingPolicyId>0</accountingPolicyId>
<accountingPolicyName>N/A</accountingPolicyName>
<accountingPolicyObjectPointer/>
<accountingOn>>false</accountingOn>
<sharedQueueOn>>false</sharedQueueOn>
<operationalFlags/>
<ingressAtmPolicyId>0</ingressAtmPolicyId>
<egressAtmPolicyId>0</egressAtmPolicyId>
<scpPointer/>
<ccagSnmpPortId>0</ccagSnmpPortId>
<mirrorStatus/>
<rowStatus>1</rowStatus>
<serviceIdString>N/A</serviceIdString>
<hsmdaEgressSecondaryShaperObjPointer/>
<hsmdaIngPackByteOff>-128</hsmdaIngPackByteOff>
<hsmdaEgrPackByteOff>-128</hsmdaEgrPackByteOff>
<ingressAggRateLimit>-1</ingressAggRateLimit>
<egressAggRateLimit>-1</egressAggRateLimit>
<egressFrameBaseAccounting>>false</egressFrameBaseAccounting>
<dosProtection>NE DoS Protection:1</dosProtection>
<dependantOlcState>inService</dependantOlcState>
<olcState>inService</olcState>
<templateVersionPointer/>
<portPointer>network:10.100.100.2:lag:interface-11</portPointer>
<innerEncapValue>125</innerEncapValue>
<outerEncapValue>11</outerEncapValue>
<portIdentifyingName>lag-11:11.125</portIdentifyingName>
<underlyingResourceState>OK</underlyingResourceState>
<compositeState>OK</compositeState>
<resourceState>OK</resourceState>
<operationalState>serviceUp</operationalState>

```

---

```

<administrativeState>serviceUp</administrativeState>
<ctpPointer>network:10.100.100.2:lag:interface-11:TP</ctpPointer>
<terminatedObjectPointer>network:10.100.100.2:lag:interface-11</terminatedObjectPointer>
<terminatedObjectId>1342177291</terminatedObjectId>
<terminatedObjectName>Lag 11</terminatedObjectName>
<routerId>1</routerId>
<routerName/>
<portId>1342177291</portId>
<portName>Lag 11</portName>
<encapType>qinqEncap</encapType>
<terminatedPortPointer>network:10.100.100.2:lag:interface-11</terminatedPortPointer>
<terminatedPortClassName>lag.Interface</terminatedPortClassName>
<terminatedPortCombinedEncapValue>8192011</terminatedPortCombinedEncapValue>
<terminatedPortInnerEncapValue>125</terminatedPortInnerEncapValue>
<terminatedPortOuterEncapValue>11</terminatedPortOuterEncapValue>
<isExclusiveTermination>>false</isExclusiveTermination>
<provisionedMtu>1650</provisionedMtu>
<actualMtu>1692</actualMtu>
<mtuMismatch>>false</mtuMismatch>
<terminationParticipationType>client</terminationParticipationType>
<domain>service</domain>
<application>serviceVpls</application>
<nodeName>SR12</nodeName>
<nodeId>10.100.100.2</nodeId>
<displayedName>Lag 11:11.125</displayedName>
<description>1234567890-1223456789</description>
<deploymentState>0</deploymentState>
<objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-outer-
tag-11</objectFullName>
<name>interface-lag_11-inner-tag-125-outer-tag-11</name>
<selfAlarmed>>false</selfAlarmed>
<children-Set>
  <vpls.L2AccessInterfaceMldSnpgCfg>
    <serviceId>1009510002</serviceId>
    <portId>1342177291</portId>
    <encapVal>8192011</encapVal>
    <importPolicy>N/A</importPolicy>
    <fastLeave>disabled</fastLeave>
    <mrouterAttached>>false</mrouterAttached>
    <sendQueries>disabled</sendQueries>
    <genQueryInterval>125</genQueryInterval>
    <queryResponseInterval>10</queryResponseInterval>
    <robustCount>2</robustCount>
    <lastMemberInterval>10</lastMemberInterval>
    <maxNbrGroups>0</maxNbrGroups>
    <mldVersion>version_2</mldVersion>
    <deploymentState>0</deploymentState>
    <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:mldSnooping</objectFullName>
    <name>mldSnooping</name>
    <selfAlarmed>>false</selfAlarmed>
    <children-Set/>
  </vpls.L2AccessInterfaceMldSnpgCfg>
  <ressubscr.SapSubMgmtCfg>
    <serviceId>1009510002</serviceId>
    <portId>1342177291</portId>
    <encapVal>8192011</encapVal>
    <neId>10.100.100.2</neId>
    <adminStatus>enabled</adminStatus>
    <serviceModel>notSpecified</serviceModel>

```

---

```

        <defaultSubProfile>Subscriber Profile:IPTV-T00</defaultSubProfile>
        <defaultSlaProfile>SLA Profile:IPTV-CLASSE00</defaultSlaProfile>
        <defaultSubIdentPolicy>Subscriber Identification Policy:IPTV-
ESM</defaultSubIdentPolicy>
        <defaultAppProfile/>
        <subscriberLimit>1</subscriberLimit>
        <profiledTrafficOnly>true</profiledTrafficOnly>
        <nonSubTrafficIdent>N/A</nonSubTrafficIdent>
        <nonSubTrafficSubProfile/>
        <nonSubTrafficSlaProfile/>
        <nonSubTrafficAppProfile/>
        <macDestAddrHashing>>false</macDestAddrHashing>
        <defSubscriberIdType>none</defSubscriberIdType>
        <defSubscriberIdString>N/A</defSubscriberIdString>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:subMgmt</objectFullName>
        <name>subMgmt</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </ressubscr.SapSubMgmtCfg>
    <vpls.L2AccessInterfaceMldMvrCfg>
        <serviceId>1009510002</serviceId>
        <siteId>10.100.100.2</siteId>
        <portId>1342177291</portId>
        <encapVal>8192011</encapVal>
        <fromVplsPointer/>
        <fromVplsId>0</fromVplsId>
        <cliCfgedFromVpls>0</cliCfgedFromVpls>
        <proxySapPointer/>
        <proxySapPortId>0</proxySapPortId>
        <proxySapEncapValue>0</proxySapEncapValue>
        <cliCfgedToSap>N/A</cliCfgedToSap>
        <fromVplsExists>true</fromVplsExists>
        <proxySapExists>true</proxySapExists>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:mldMvr</objectFullName>
        <name>mldMvr</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </vpls.L2AccessInterfaceMldMvrCfg>
    <antispoof.L2AntiSpoofing>
        <serviceId>1009510002</serviceId>
        <portId>1342177291</portId>
        <encapVal>8192011</encapVal>
        <antiSpoofing>sourceIpAndMacAddr</antiSpoofing>
        <arpReplyAgent>enabled</arpReplyAgent>
        <macPinning>enabled</macPinning>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:antiSpoofing</objectFullName>
        <name>antiSpoofing</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </antispoof.L2AntiSpoofing>
    <12fwd.AccessInterfaceFib>
        <entries>0</entries>
        <staticEntries>0</staticEntries>
        <agingEnabled>true</agingEnabled>

```

---

```

    <learningEnabled>true</learningEnabled>
    <maxEntries>3</maxEntries>
    <limitMacMove>blockable</limitMacMove>
    <discardUnknownSource>>false</discardUnknownSource>
    <macMoveRateExcdLeft>4</macMoveRateExcdLeft>
    <macMoveNextUpTime>0</macMoveNextUpTime>
    <limitMacMoveLevel>tertiary</limitMacMoveLevel>
    <restrictProtectedSource>>false</restrictProtectedSource>
    <restrictUnprotectedDestination>>false</restrictUnprotectedDestination>
    <restrictProtectedSourceAction>disable</restrictProtectedSourceAction>
    <portPointer>network:10.100.100.2:lag:interface-11</portPointer>
    <portId>1342177291</portId>
    <combinedEncapValue>8192011</combinedEncapValue>
    <innerEncapValue>125</innerEncapValue>
    <outerEncapValue>11</outerEncapValue>
    <portName>Lag 11</portName>
    <serviceId>1009510002</serviceId>
    <svcComponentId>1431</svcComponentId>
    <serviceName>VPLS 1009510002</serviceName>
    <subscriberId>5000</subscriberId>
    <subscriberName>IPTV</subscriberName>
    <siteId>10.100.100.2</siteId>
    <siteName>SR12</siteName>
    <deploymentState>0</deploymentState>
    <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:fib</objectFullName>
    <name>fib</name>
    <selfAlarmed>>false</selfAlarmed>
    <children-Set/>
  </l2fwd.AccessInterfaceFib>
  <vpls.L2AccessInterfaceMvrCfg>
    <serviceId>1009510002</serviceId>
    <siteId>10.100.100.2</siteId>
    <portId>1342177291</portId>
    <encapVal>8192011</encapVal>
    <fromVplsPointer/>
    <fromVplsId>0</fromVplsId>
    <cliCfgedFromVpls>0</cliCfgedFromVpls>
    <proxySapPointer/>
    <proxySapPortId>0</proxySapPortId>
    <proxySapEncapValue>0</proxySapEncapValue>
    <cliCfgedToSap>N/A</cliCfgedToSap>
    <fromVplsExists>true</fromVplsExists>
    <proxySapExists>true</proxySapExists>
    <deploymentState>0</deploymentState>
    <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:mvr</objectFullName>
    <name>mvr</name>
    <selfAlarmed>>false</selfAlarmed>
    <children-Set/>
  </vpls.L2AccessInterfaceMvrCfg>
  <vpls.L2AccessItdhcpRelayCfg>
    <serviceId>1009510002</serviceId>
    <portId>1342177291</portId>
    <encapVal>8192011</encapVal>
    <administrativeState>enabled</administrativeState>
    <description>N/A</description>
    <snooping>enabled</snooping>
    <leasePopulate>3</leasePopulate>
    <operLeasePopulate>0</operLeasePopulate>

```

---

```

        <infoAction>keep</infoAction>
        <infoCircuitId>asciiTuple</infoCircuitId>
        <infoRemoteId>>false</infoRemoteId>
        <remoteIdString>N/A</remoteIdString>
        <proxyAdminState>>false</proxyAdminState>
        <emulatedServerAddr>0.0.0.0</emulatedServerAddr>
        <leaseTime>disabled</leaseTime>
        <proxyLeaseTime>0</proxyLeaseTime>
        <leaseTimeDays>0</leaseTimeDays>
        <leaseTimeHours>0</leaseTimeHours>
        <leaseTimeMinutes>0</leaseTimeMinutes>
        <leaseTimeSeconds>0</leaseTimeSeconds>
        <proxyLTRadiusOverride>>false</proxyLTRadiusOverride>
        <vendorIncludeOptions/>
        <vendorOptionString>N/A</vendorOptionString>
        <subscrAuthPolicyPointer/>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:dhcp</objectFullName>
        <name>dhcp</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </vpls.L2AccessItfDhcpRelayCfg>
    <vpls.L2AccessInterfaceIgmppSnpGCfg>
        <serviceId>1009510002</serviceId>
        <portId>1342177291</portId>
        <encapVal>8192011</encapVal>
        <importPolicy>N/A</importPolicy>
        <fastLeave>disabled</fastLeave>
        <mrouterAttached>>false</mrouterAttached>
        <sendQueries>disabled</sendQueries>
        <genQueryInterval>125</genQueryInterval>
        <queryResponseInterval>10</queryResponseInterval>
        <robustCount>2</robustCount>
        <lastMemberInterval>10</lastMemberInterval>
        <maxNbrGroups>0</maxNbrGroups>
        <igmpVersion>version_3</igmpVersion>
        <mCastCacPolicyName>N/A</mCastCacPolicyName>
        <mCastCacPolicyPointer/>
        <unconstrainedBandwidth>-1</unconstrainedBandwidth>
        <preRsvdMandatoryBandwidth>-1</preRsvdMandatoryBandwidth>
        <mCacConstAdminState>tmnxInService</mCacConstAdminState>
        <inUseMandatoryBw>0</inUseMandatoryBw>
        <inUseOptionalBw>0</inUseOptionalBw>
        <availableMandatoryBw>4294967295</availableMandatoryBw>
        <availableOptionalBw>4294967295</availableOptionalBw>
        <bwValuesInTransit>>false</bwValuesInTransit>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:igmpSnooping</objectFullName>
        <name>igmpSnooping</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </vpls.L2AccessInterfaceIgmppSnpGCfg>
    <service.SapTodMonitor>
        <serviceId>1009510002</serviceId>
        <portId>1342177291</portId>
        <encapValue>8192011</encapValue>
        <currentIngressIpFilterID>0</currentIngressIpFilterID>
        <currentIngressIpV6FilterID>0</currentIngressIpV6FilterID>

```

---

```

        <currentIngressMacFilterID>0</currentIngressMacFilterID>
        <currentIngressQosPolicyID>1</currentIngressQosPolicyID>
        <currentIngressQosSchedulerName>N/A</currentIngressQosSchedulerName>
        <currentEgressIpFilterID>0</currentEgressIpFilterID>
        <currentEgressIpV6FilterID>0</currentEgressIpV6FilterID>
        <currentEgressMacFilterID>0</currentEgressMacFilterID>
        <currentEgressQosPolicyID>37000</currentEgressQosPolicyID>
        <currentEgressQosSchedulerName>N/A</currentEgressQosSchedulerName>
        <intendedIngressIpFilterID>0</intendedIngressIpFilterID>
        <intendedIngressIpV6FilterID>0</intendedIngressIpV6FilterID>
        <intendedIngressMacFilterID>0</intendedIngressMacFilterID>
        <intendedIngressQosPolicyID>1</intendedIngressQosPolicyID>
        <intendedIngressQosSchedulerName>N/A</intendedIngressQosSchedulerName>
        <intendedEgressIpFilterID>0</intendedEgressIpFilterID>
        <intendedEgressIpV6FilterID>0</intendedEgressIpV6FilterID>
        <intendedEgressMacFilterID>0</intendedEgressMacFilterID>
        <intendedEgressQosPolicyID>37000</intendedEgressQosPolicyID>
        <intendedEgressQosSchedulerName>N/A</intendedEgressQosSchedulerName>
        <deploymentState>0</deploymentState>
        <objectFullName>svc-mgr:service-1431:10.100.100.2:interface-lag_11-inner-tag-125-
outer-tag-11:svc-1009510002port-1342177291encap-8192011</objectFullName>
        <name>svc-1009510002port-1342177291encap-8192011</name>
        <selfAlarmed>>false</selfAlarmed>
        <children-Set/>
    </service.SapTodMonitor>
</children-Set>
</vpls.L2AccessInterface>
</data>
</lrResponse>
</lrResponseList>
</rfsResponse>

```



# Referências Bibliográficas

- [1] NGOSS Solution Frameworks. [Online] 2008.  
<http://www.tmforum.org/pages/1911/default.aspx>.
- [2] TeleManagement Forum. [Online] 2008. [www.tmforum.org](http://www.tmforum.org).
- [3] International Telecommunication Union (ITU). [Online] 2008.  
<http://www.itu.int/net/about/landmarks.aspx>.
- [4] International Telecommunication Union – Telecommunications (ITU-T). [Online] 2008. <http://www.itu.int/ITU-T/>.
- [5] Internet Engineering Task Force (IETF). [Online] 2008. <http://www.ietf.org/>.
- [6] Sexton, Mike e Reid, Andy. *Broadband Networking: ATM, SDH, and SONET*. s.l. : Artech House, 1997.
- [7] Yamanaka, Naoaki, Shiimoto, Kohei e Oki, Eiji. *GMPLS Technologies: Broadband Backbone Networks and Systems*. s.l. : Taylor & Francis Group, 2005.
- [8] RFC 791. [Online] 2008. <http://www.rfc-editor.org/rfc/rfc791.txt>.
- [9] Tanenbaum, Andrew S. *Computer Networks, Third Edition*. s.l. : Prentice Hall.
- [10] Black, U. *MPLS and label switching networks*. s.l. : Prentice Hall, 2001.
- [11] Stallings, W. *Data and computer communications, 5th Edition*. s.l. : Prentice-Hall, 1997.
- [12] *El mundo de los OSS y BSS ante las nuevas exigencias de negocio*. Hita, Carmen de, Lozano, José Antonio e Morilla, Julio. s.l. : Comunicaciones de Telefónica I+D, 2004, Vol. 33.
- [13] Iniciativa OSS/J. [Online] 2008. [www.ossj.org](http://www.ossj.org).
- [14] OSS through Java as an Implementation of NGOSS. s.l. : White Paper, 2004.
- [15] *NGOSS Rationale, benefits & current status*. Creaner, Martin. s.l. : OSS Forum, 2004.
- [16] *COTS-Based Software Systems: 4th International Conference*. s.l. : Bilbao, Spain, 2005.
- [17] *Update on NGOSS Contract*. Covino, G. e Hardwike, Jim. s.l. : TeleManagementWorld, 2004.

- [18] *Enhanced Telecom Operations Map (eTOM), The Business Process Framework*. 2004, Vol. TMF Approved Version 4.0.
- [19] *Shared Information/Data (SID) Model, Concepts, Principles and Domains*. s.l. : TMF Member Evaluation, 2004, Vol. Release 4.5.
- [20] The British Telecom. [Online] 2008. <http://www.bt.com/>.
- [21] Case Study-EAI in NGOSS: A Migration for an IP OSS. [Online] 2008. <http://www.tmforum.org/page28983.aspx>.
- [22] Sun microsystems. [Online] 2008. <http://www.sun.com/>.
- [23] Who makes what: OSS. [Online] 2004. [www.nextgenerationservices.com](http://www.nextgenerationservices.com).
- [24] OSS Market Perception Study. [Online] 2003. [www.heavyreading.com](http://www.heavyreading.com).
- [25] *NGOSS in practice: examining the success of implementing the TMF NGOSS framework*. Covino, Giuseppe. s.l. : Global OSS Forum, 2004.
- [26] *COVAD. Applying Commercial Applications as a Strategy for a Turbulent Business*. Grantham, Paul. s.l. : Global OSS Forum, 2004.
- [27] *The Future of IP Services*. Howard, Greg. s.l. : HTRC Group.
- [28] PT Inovação, S.A. [Online] 2008. <http://www.ptinovacao.pt/>.
- [29] *Ativação Automática de Serviços - Saber e Fazer Telecomunicações*. Gonçalves, Jorge, et al. s.l. : PTInovação, 2005, Vol. 3.
- [30] Weiskirchner, Heinz. *Telecommunications management networks (TMN): introduction and usage within broadband transmission networks*. s.l. : R. Oldenbourg Verlag GmbH, 1993.
- [31] Fernandes, José Pedro, et al. *GRPG II - Propostas / Recomendações de Evolução*. s.l. : PTInovação, 2003.
- [32] Engenharia de Requisitos. [Online] 2008. [http://dein.ucs.br/profs/gerocco/engRequisitos\(conceitos\).pdf](http://dein.ucs.br/profs/gerocco/engRequisitos(conceitos).pdf).
- [33] Silva, Nuno, et al. *Análise das tecnologias emergentes para a integração de processos e sistemas*. PTInovação, 2004.
- [34] JBoss. [Online] 2008. <http://www.jboss.org/>.
- [35] Java EE at a Glance. [Online] 2008. <http://java.sun.com/javae/>.
- [36] *OSS/J & NGOSS Part II: The New Realities*. Bauerfeld, Wulf, Frankenberger, Joerg e Holmes, Phil. s.l. : TeleManagementWorld, 2004.
- [37] WebServices Axis [Online] 2008. <http://ws.apache.org/axis/>.



